CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE

Canada

NL 339 (r. 86/08)

145

0-315-21339-6

National Library of Canada
Canadian Theses Division
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Division des thèses canadiennes

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

Ronald Collette

Date of Birth — Date de naissance

31 MARS 1985

Country of Birth — Lieu de naissance

MONTRÉAL

Permanent Address — Résidence fixe

855 CHARLES ROY
BOUCHERUILLE
QUÉBEC

Title of Thesis — Titre de la thèse

A three-dimensional model for Simulating Flow and Mass Transport in Groundwater Systems

University — Université

University of Alberta

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M. Sc

Year this degree conferred — Année d'obtention de ce grade

1985

Name of Supervisor — Nom du directeur de thèse

F.W. Schwartz

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

January 25 1985

Signature

Ronald Collette

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

A THREE-DIMENSIONAL MODEL FOR SIMULATING FLOW AND MASS TRANSPORT IN GROUNDWATER SYSTEMS

by

© Ronald Collette

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCES

DEPARTMENT OF GEOLOGY

EDMONTON, ALBERTA

SPRING 1985

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Ronald Collette

TITLE OF THESIS A THREE-DIMENSIONAL MODEL FOR SIMULATING FLOW AND MASS TRANSPORT IN GROUNDWATER SYSTEMS

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCES

YEAR THIS DEGREE GRANTED SPRING 1985

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Ronald Collette

PERMANENT ADDRESS:

855 Charles Roy

Bacherville (Montréal)

Québec

DATED January 25 1985

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A THREE-DIMENSIONAL MODEL FOR SIMULATING FLOW AND MASS TRANSPORT IN GROUNDWATER SYSTEMS submitted by Ronald Collette in partial fulfilment of the requirements for the degree of MASTER OF SCIENCES.

Frank W. Schwartz

Supervisor

Date..January..25.....1985.....

ABSTRACT

During the past ten years the scope of modeling activity has broadened with the development of predictive models for mass transport in groundwater systems. Various shortcomings of the finite-difference and finite-element techniques used for the solutions to the mass-transport equation have resulted in the development of direct-simulation techniques such as the Discrete-Parcel-Random-Walk method (DPRW). The development of computer codes for three-dimensional applications has so far received limited attention because of the complex data handling, computing costs and storage limitations commonly involved.

A three-dimensional flow and transport model (3DFT) has been developed which combines a numerical solution to the flow equation, using a finite-element technique, with a direct-simulation of mass transport by means of a DPRW technique. The computer code uses a mesh generator that considerably reduces the input required and a simple input format by means of numerical maps. It can handle a great variety of boundary conditions for steady-state or transient-state flow simulations. Once the distribution of hydraulic head is obtained, the transport simulation takes place by inserting reference particles in the flow field at rectangular source-areas along any boundary. Particles are displaced along flow vectors in order to simulate advection. Dispersion is accounted for by adding a random component to the deterministic motion. Particle motion is restricted by

the flow-domain's boundaries and by specified sinks.

The extreme structural sparseness of the finite-element matrices allows for a substantial reduction in memory by using a compressed storage mode combined to a sparse-matrix solver. This storage gain is compensated by the requirement of a large workspace for the direct solution of the matrix system resulting in limitations in grid size but the cost efficiency is considerably improved.

The accuracy of the code has been tested for steady-state and transient-state flow conditions with good results. The comparison of transport simulations with analytical results shows the averaging effect of the particle-tracking method. The few parameters critical for the accuracy of the solution are the transport time-steps, the element size and the total number of particles used. With a proper choice of parameters, numerical results compare favorably with the analytical solutions for one and three-dimensional transport systems.

ACKNOWLEDGMENTS / REMERCIEMENTS

I am grateful to professor F.W. Schwartz who made this work possible and easier by providing a constant support and by reviewing it throughout its completion. Thanks also to the members of my committee; N.W. Rutter, K. Muehlenbachs and J.D. Scott for reviewing the thesis.

Sincere thanks go to Allan Crowe who taught me the basics of model development and computing at the U of A and to the Department of Geology's computer expert Desmond Wynne for many helpful insights and priceless technical advices. I am also indebted to W.A. "Literature King" Milne-Home for precious information and accurate references given on repeated occasions.

Enfin je tiens à remercier de façon toute spéciale celle qui m'a été proche au cours de cette entreprise, Lise Allard, et avec qui j'ai surmonté tant de difficultés.

Table of Contents

Chapter — Page

I. INTRODUCTION 1

A. Review of Previous Work and Statement of Objectives 1

B. General Approach and Organization of the Study .. 8

II. MODELING METHODOLOGY 12

A. Theoretical Overview 12

B. Design of the Computer Program 15

Data Input, Mesh Generation and Boundary Conditions 15

Solving for Flow and Mass Transport 21

III. EVALUATION OF THE MODEL 34

A. Storage Requirements and Computer Effort 34

B. Verification of the Flow Simulation Results 41

Steady-State Flow 41

Transient-State Flow 47

C. Verification of the Mass-Transport Simulations . 51

One-dimensional Simulation 53

Three-dimensional Simulations 57

D. Practical Constraints in Applying the Code 69

IV. CONCLUSIONS AND RECOMMENDATIONS 72

BIBLIOGRAPHY 76

APPENDIX 1. 3DFT User's Manual 81

List of Figures

Figure		Page
I.1	Flowchart outlining the methodology used in the development of the computer code	9
II.1	Generalized flowchart of the 3-D Flow and Transport code	16
II.2	Figure showing the three-dimensional grid and example of a typical element	18
II.3	Modifications to the Flow Domain	19
III.1	Illustration of the core requirement for various program segments	35
III.2	A comparison of the relative computer effort for each segment	40
III.3	Figure showing the setup and results of a simulation of flow in a simple rectangular domain	42
III.4	Illustration of the "island discharge" problem	44
III.5	Time-drawdown comparison of theoretical and digital computer solutions	49
III.6	The particle distribution in each element of the grid resulting from the instantaneous injection of 32000 particles	52
III.7	Numerical map of mass concentration 16.8 years after the instanteneous input of mass at a point source	54
III.8	The results from 3DFT compared with the analytical solution to the one-dimensional advection diffusion equation	56
III.9	A given injection rate C_oQ can be obtained by a point source or by an equivalent area source	61
III.10	This figure depicts the plume shape in two horizontal planes at two depths and three different times	62

Figure Page

III.11 The numerical solution is obtained by dividing the total particle mass in any element by its volume 64

III.12 The lateral spreading of the contaminant is compared with an averaged analytical solution 66

III.13 The sensitivity of the numerical solution to the discrete character of the source function 68

List of Tables

Table		Page
III.1	Comparison of the efficiency between the dense and the sparse solver	38

I. INTRODUCTION

A. Review of Previous Work and Statement of Objectives

Quantitative methods have been used in groundwater geology since the early 1900's (Thiem, 1906). Early applications of these methods became possible as soon as empirical laws and systematic observations (Darcy, 1856; Chamberlin, 1885) were formulated in the proper mathematical and conceptual framework (Forscheimer, 1886, 1898; Slichter, 1898 in Freeze et al., 1983).

Many of the early modeling efforts in hydrogeology were concerned with basic problems related to groundwater resource evaluation. However, during the past ten years, the scope of modeling activity has broadened with the development of predictive models for mass transport in groundwater systems. Such models have been applied to a variety of chemical and contaminant problems related to the infiltration of tailing leachates, seawater intrusion, chemical spills and sanitary landfills.

Mathematical models for physico-chemical transport in groundwater systems can be formulated from two different viewpoints, namely a "model-equation" approach and a "direct-simulation" approach (Ahlstrom et al., 1977). The first approach involves the abstraction of processes by one or more partial differential equations and controlling parameters, which can be solved using several well-known numerical techniques such as finite difference and finite

element.

Examples of the development and application of the finite-difference method are provided by the work of Shamir and Harleman (1967), Lawson (1971) and Dillon et al. (1978). Finite-element solutions have been used by Guymon et al. (1970), Pinder (1973), Segol (1977) and Gray and Hoffmann (1983).

These numerical approaches are most successful if the process of mass transport is dispersion-dominated in which case the transport equation shows a parabolic behavior (Neuman, 1979). They do however suffer from problems of numerical dispersion and instability in advection-dominated systems. For those systems, the differential equation becomes hyperbolic in character and hence much more difficult to solve numerically. Because many transport problems are of the advection-dominated type, other solution techniques have been developed. These approaches are typically less rigorous and are based on techniques of direct simulation that have been developed specifically to deal with problems inherent with the model-equation approaches.

Techniques of direct simulation involves to the use of numerical features, like reference particles and velocity fields, which are allowed to interact as determined by the physical constraints of the simulated system (Ahlstrom et al., 1977). The most common of hyperbolic approaches are the method of characteristics (MOC) (Gardner et al., 1964;

Pinder and Cooper, 1970 and Bredehoeft and Pinder, 1973) and the discrete-parcel-random-walk (DPRW) (Ahlstrom et al., 1977 and Schwartz and Crowe 1980).

With the MOC method the transport equation is not solved directly. It is replaced by an equivalent set of ordinary differential equations by using moving fluid particles instead of fixed points in space as references. Moving fluid particles thus simulate convective transport. Changes in concentration due to dispersion and other phenomena are evaluated using a finite-difference approximation to the remaining differential equations. The DPRW method similarly describes the spread of a large number of reference particles in the flow field. However, particles are associated to a given mass and dispersion is simulated by the redistribution of particles around the average flow path according to the dispersive character of the porous media.

The MOC is relatively simple in concept and overcomes the numerical problems associated to the finite-difference and finite-element approaches. However, it has proven to be tedious to program (Pinder, 1973) and suffers from various shortcomings, namely in its treatment of sources and sinks, converging flow regimes and mass balance (Neuman, 1979). Stability also poses limitations in term of time step size.

The DPRW like MOC is based in part on tracking reference particles within a domain. However unlike the MOC the DPRW approach is always mass conservative, does not

exibit any cumulative numerical dispersion and is inherently stable. It permits considerable flexibility in the handling of source and sink terms and boundary conditions. The overall simplicity of the DPRW procedure in addition makes it easier to use and implement in a computer code. The main drawback of this approach concerns computational speed and accuracy.

Direct numerical approaches are usually faster than DPRW especially if a high degree of accuracy is needed. For example, numerical experiments have shown that relatively large numbers of reference particles must be included in the solution to obtain reasonable accuracy. Typically the total number of particles must be multiplied by four in order to double the accuracy of the results (Ahlstrom et al., 1977). A DPRW solution always include a certain amount of noise because of its statistical nature. Also, the absence of straightforward criteria concerning the minimal number of particles needed to achieve any given degree of accuracy requires that a user must experiment with particle densities when dealing with specific problems.

Notwithstanding these shortcomings, a number of features specific to the DPRW makes it worth further development. Because it requires only limited addressable memory at any time it is amenable to implementation on smaller and usually more cost-effective computer systems. The validity of any simulation can easily be tested by using only a few control particles for a preliminary debug run.

Subsequent runs can also be averaged for increased accuracy. Finally DPRW formulation appears to be reasonnably well suited for incorporating chemical reactions and ion exchange processes (Ahlstrom et al., 1977). Such a method is generally much less complex than the model-equation and considerably easier to develop and implement.

A variety of practical applications (Prickett et al., 1981; Schwartz, 1977 and Schwartz and Donath, 1980) have demonstrated the application of DPRW methods in two-dimensional models. For a limited number of geological settings, which can be approximated well in two dimensions, these models have been very useful. However for many problems, a two-dimensional description is simply not adequate to represent the complex features of groundwater flow and transport. The inherent three-dimensionality of these groundwater flow systems requires a detailed description of variability in all coordinate directions because the interaction of all the components of flow cannot be properly averaged in any simple way. The spatial averaging also gives rise to the scaling-up of dispersivity coefficients as a result of calibrating a two-dimensional model on a three-dimensional system (Domenico and Robbins, 1984a).

A limited number of studies (Ahlstrom et al., 1977; Dillon et al., 1978, and Segol 1977) have been concerned with treating the transport numerically in three dimensions. Segol's (1977) finite-element code treats saturated-

unsaturated flow and mass transport. Indications are that the code is expensive to run. The very fine grid generally required makes it impractical for fully three-dimensional field situations. Ahlstrom et al. (1977) formulated the DPRW procedure for applications in two and three dimensions. The code that was developed is only two-dimensional and does not include a solution for the groundwater-flow equation on which particle displacement is dependent. Data necessary to describe the velocity field thus must be provided beforehand.

Dillon et al. (1978) describe a functional three-dimensional mass transport model also capable of simulating heat transfer and the decay of radionuclide chain. The program, Sandia Waste Isolation Flow and Transport (SWIFT), has been used mainly for the evaluation of potential repositories for nuclear wastes. The code uses a finite-difference formulation to solve the coupled heat, mass and flow equations. It probably represents the most comprehensive code developed to date. The finite-difference solution to the mass-transport problem is however subject to the significant numerical dispersion, which often occurs with the finite-difference approach. The major consequences of this problem are constraints on either the grid block or time step size or both. Finally the generality of the code and its comprehensive nature puts a considerable burden on the users capability to handle its complex input requirements.

One important general limitation of existing three-dimentional models is related to the prohibitive computer effort and excessive storage requirements needed to treat even moderately sized problems. Most do not make use of any technique that can minimize the costs of computer processing. In addition, because these codes have limited availability and have ackward input requirements, they are not used generally in field applications. In fact extensive modifications to the programs and elaborations of the documentation is generally required to make these models useful for the general user (Chorley et al., 1982).

The purpose of this study is to develop and verify a practical computer model capable of making three-dimensional mass transport simulations amenable for the general user. The computer code solves for flow and mass transport independently in an uncoupled manner so that there is no need to iterate for convergence. Implicit in this approach is the assumption that the changes in concentration do not affect the flow field. The flow equation is solved using a finite-element procedure as it allows for a flexible grid design and yields a good approximation to space derivatives (Pinder and Gray, 1977). The solution to flow is provided by an original three-dimensional code. Time-dependent solutions are computed by a straightforward finite-difference interpolation. A new particle-tracking code is developed in this study to simulate solute transport. It is basically an extension of the existing concepts actually applied to

two-dimensional works. As will become apparent, a variety of different techniques has been used to simplify the specification of input data to the code.

Storage requirements and execution time which are bound to be large for three-dimensional problems have received special attention in this study. A sparse-matrix solver is used in conjunction with a highly compressed storage mode. The computer code itself contains features to help debugging and testing. A set of execution messages keeps the user informed of the successful completion of operations. Intermediate computational results can be displayed to provide control and detailed testing. Finally a fully documented user guide is designed to assist in the implementation of the program and the application of the code in practice.

B. General Approach and Organization of the Study

The general approach to model development, as shown in Figure I.1, involves five basic steps. First, the physical system of interest must be conceived in terms of basic processes and measurable parameters. In the case of the subsurface transport of mass the process involves groundwater flow and physico-chemical transport. The dependent variables are the hydraulic head and solute concentration. Having characterized the problem conceptually, the next step involves expressing these concepts in mathematical terms. This can be done by

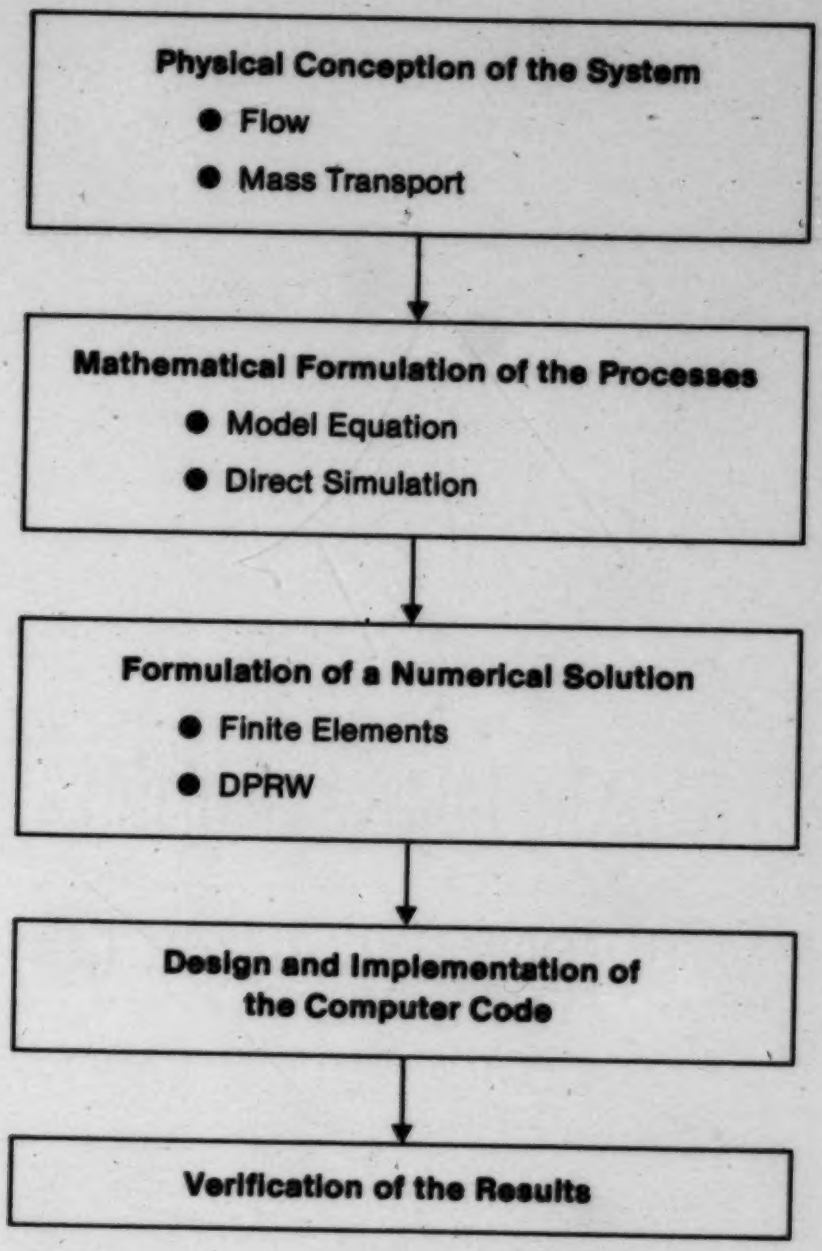


Figure I.1
Flowchart outlining the methodology used in the development of the computer code. Adapted from Carey and Oden (1984) page 3.

formulating the governing mathematical equations for the model-equation approach (Mercer and Faust, 1981) and by setting the interacting structures for the direct-simulation approach. The groundwater flow equation, of the partial differential type, describes the array of processes that operate to control groundwater seepage. When provided with the proper constraints in term of initial and boundary conditions, it can be solved to obtain the distribution, over space or time, of the hydraulic heads.

The third step is to formulate a numerical solution to the problem. The general solution to partial differential equations is provided by numerical methods (Mercer and Faust, 1981). They consist of a discrete approximation of the original continuous equations. The mathematical problem eventually is normally reduced to solving a set of simpler algebraic equations. Once the distribution of heads over the domain is known, the velocity field can be determined and the solution to mass transport can be obtained directly by means of DPRW method. The new set of equations and the particle-tracking technique are amenable to a final solution by means of digital computer processing as a fourth step. For a limited set of constraints exact analytical solutions to flow and mass transport processes are known. At the final step, the validity of the model can be verified by comparing the numerical approximations to these mathematical solutions.

This generalized approach to the formulation of the mathematical model here forms the basis for organizing this thesis. Thus chapter two focuses on the first four steps, namely on the problems of conceptualizing the solute-transport problem, its mathematical formulation, and the detailed numerical approaches adopted as well as their implementation in a computer code. Chapter three describes the steps taken to verify the model.

II. MODELING METHODOLOGY

A. Theoretical Overview

The transport of dissolved contaminants in a groundwater-flow system is governed primarily by the physical processes of advection and dispersion and a variety of chemical processes. Advection refers to the migration of a dissolved mass by the motion of the water. In most cases the velocity of solute migration is the same velocity as the groundwater. Dispersion refers to the irreversible mixing of a solution with surrounding groundwaters causing a progressive dilution. A variety of chemical processes can affect contaminant distribution in groundwater systems. Adsorbtion, ion exchange, biological degradation and radioactive decay are often recognized to have an impact on the concentration patterns of many contaminants. Although these chemical and biological processes are closely related to problems of water quality, the present modeling effort is concerned with the physical aspects of mass transport.

The transport of a single, dissolved contaminant which is assumed to enter a groundwater system at some source is simulated by using a Discrete-Parcel-Random-Walk (DPRW) approach (Ahlstrom et al., 1977; Schwartz and Crowe, 1980). This approach in general does not provide a direct numerical solution to the advection-dispersion equation. It is rather a direct-simulation approach based on the assumption that dissolved material can be represented as an ensemble of

discrete particles of matter. The DPRW method deals with the problem of describing the spread of a large number of moving reference particles within a flow domain. The deterministic part of the particle motion accounts for advection and the probabilistic part accounts for dispersion. The space varying distribution of the particle swarm provides a solution which is comparable to those obtained from the differential equation (Schwartz, 1984).

In order to define particle motion caused by the process of advection, it is necessary to define the groundwater velocity field. In most real cases, this field is not known *a priori* and is determined by solving a groundwater flow equation. The following equation describes the distribution of hydraulic potential or head in a three-dimensional anisotropic and heterogeneous porous medium:

$$\frac{\partial}{\partial x_i}\left[K_{ij}\frac{\partial h}{\partial x_j}\right] = S_s\frac{\partial h}{\partial t} + Q \qquad (1)$$

where: h = hydraulic head, (L);

K_{ij} = hydraulic conductivity tensor, (L/T);

S_s = elastic specific storage, (L^{-1});

Q = rate of recharge or discharge, (T^{-1});

i,j = 1,2,3 coordinate direction.

This differential equation can be solved for a set of boundary and initial conditions. The head and conductivity values are substituted into the Darcy equation to calculate

velocities anywhere in the flow domain and the result is divided by the local porosity value to obtain the seepage velocity.

To account for dispersion the rules of probability are applied to the particle motion. In other words, the reference particles are displaced from the position they would occupy due to advection alone. Thus, the swarm of reference particles is moved in part deterministically in response to advection and in part probabilistically in response to dispersion. Both of these displacements are determined by seepage velocities and in the case of dispersion from dispersivity values as well. (Schwartz, 1977).

The deterministic-probabilistic approach to modeling mass transport has been investigated and used by Ahlstrom et al. (1977), Schwartz (1977), Schwartz and Smith (1981), Smith and Schwartz (1980, 1981) and Prickett et al. (1981). It has proven helpful in investigating groundwater contamination problems and represents a valid investigation tool for applied research. The main success of this approach is related to the combination of an inherent simplicity of the method itself and the overall flexibility, which makes the approach applicable to a great variety of theoretical and practical flow and transport conditions.

B. Design of the Computer Program

This section explains how the particle-tracking technique described above has been developed as a general three-dimensional flow and transport code (3DFT). The program is implemented in FORTRAN IV on the University of Alberta's Amdahl 5860 mainframe. The computer code is developed in a structured manner with a number of relatively simple subprograms, each subdivided into distinct logical blocks. The basic sequence of operations, as shown in FIGURE II.1, involves five steps: (1) reading the input data and options, and providing the user with a data echo, (2) generating a three-dimensional mesh, (3) integrating the flow equation with a finite-element method, (4) solving the resulting system of linear equations and finally (5) solving the mass transport problem using a particle-tracking technique (DPRW).

Data Input, Mesh Generation and Boundary Conditions

Several user-oriented features have been built into the data entry procedures to help simplify the operation of the program. The most important features include (1) the output of a complete echo of the input data, (2) the capability of generating a finite-element mesh and (3) the provision of options for controlling program execution and data output. This last set of options allows the user to run and test parts of the program independently and to assist in the early stages of implementation. In addition, options for

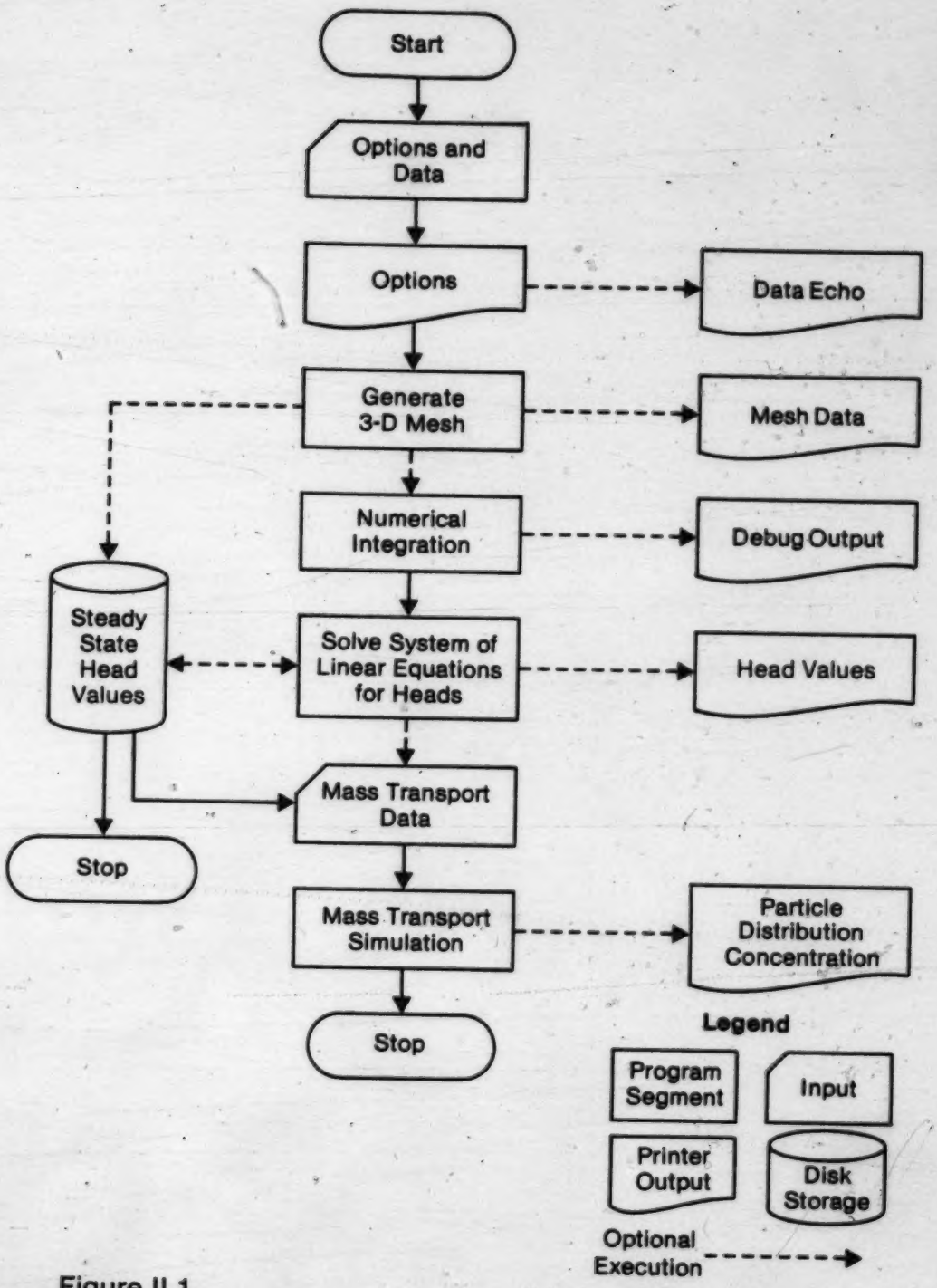


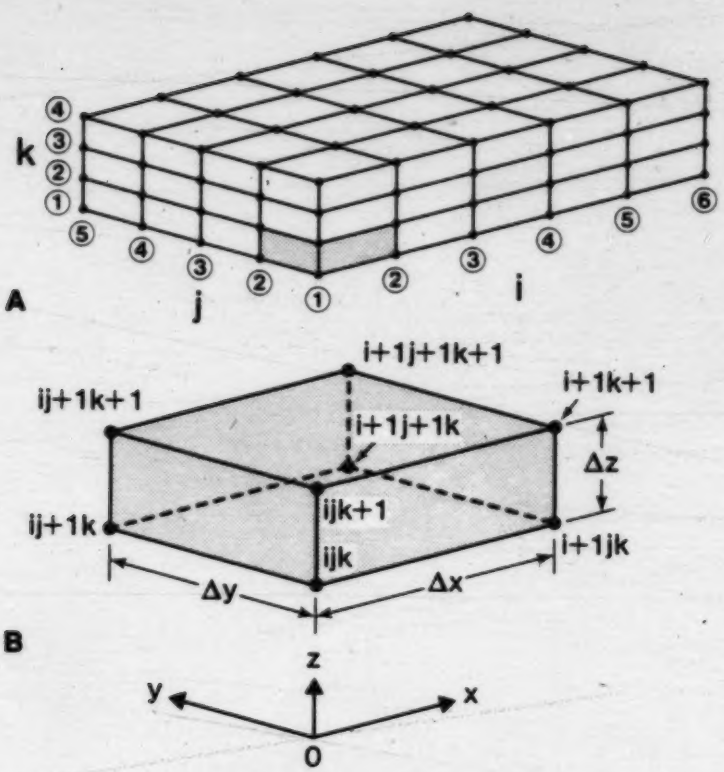
Figure II.1
Generalized flowchart of the 3-D Flow and Transport code (3DFT) showing the essential functions and program segments.

echoing the input data permits the display of numerical information at each level of processing.

Mesh generation is fully automatic given the grid size in all three directions in terms of rows, columns and layers of nodes in addition to the element length, width and height (see Figure II.2) Using the program to generate appropriate nodal coordinates, node numbers, element incidences and element numbers arrangement saves the user the tedious procedure of specifying these parameters as input. It is assumed that the finite-element grid has its greatest elongation along the i axis and its shortest along the k axis with the j axis being intermediate in size: $i \geq j \geq k$. The nodes are numbered along the shortest axis in order to minimize the bandwidth. The grid is an assemblage of brick-shaped elements with linear edges (i.e. two nodes per edge).

A simple grid having an even thickness, is illustrated in Figure II.2. Such a grid is useful to simulate a great number of confined aquifer situations, but for some simulations where irregular boundaries are encountered, the shape of the flow domain can be modified by varying the thickness of the grid (Figure II.3). In these cases, a significant economy, in term of computing effort, generally results.

The computation of hydraulic heads in any given hydrogeological system requires the specification of boundary conditions. For most cases, these boundaries can be



Coordinate system

• node
i=1, NODX nodal position along the x-axis,
j=1, NODY nodal position along the y-axis,
k=1, NODZ nodal position along the z-axis,
NODX, NODY, NODZ = number the rows, colums and layers of nodes respectively.

Δx element length
Δy element width
Δz element height

Figure II.2
A Figure showing the three-dimensional grid as an assembly of brick-shaped elements with the node positions. The grid has 6X5X4 nodes.
B Example of a typical element, its 8 surrounding nodes and their relative position in the grid together with the specified distance between nodes (ΔX, ΔY and ΔZ).

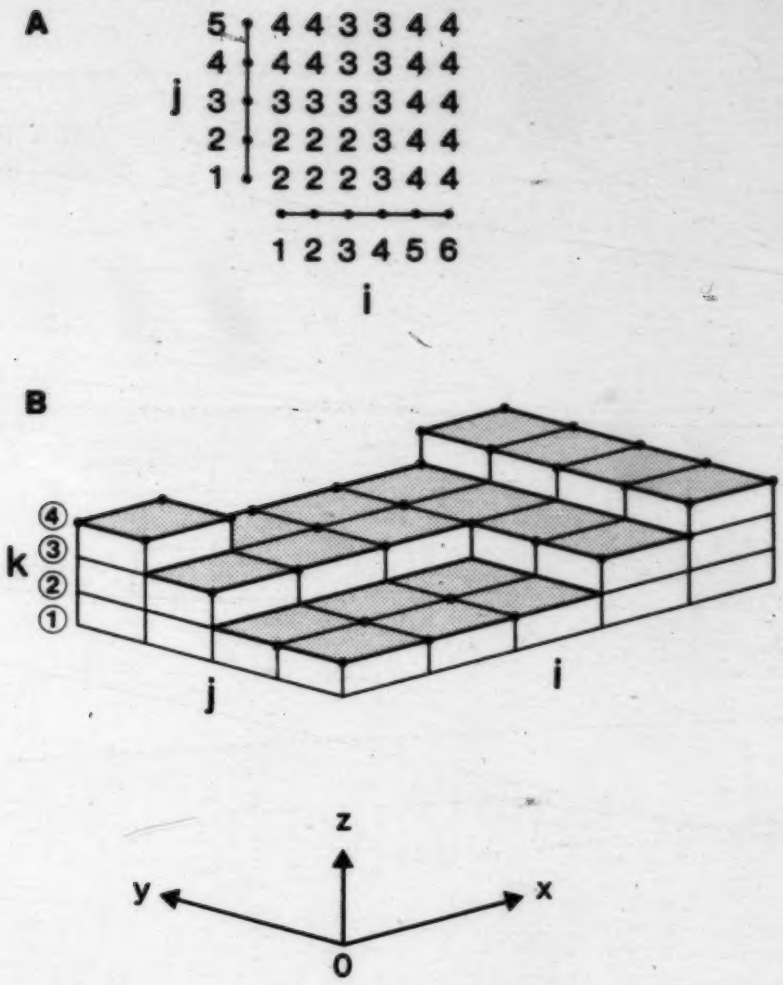


Figure II.3
Figure showing how an irregular water table configuration is represented in the input data.

A The water table elevation (IDWT) is specified for each top node with $2 \leq IDWT \leq NODZ$.

B The variable thickness results in the removal of some of the top nodes and the corresponding elements. In this case $1 \leq IDWT \leq 4$.

specified as constant-head, no-flow or specified-flow boundaries. Constant-head boundaries commonly refer to boundaries along which the hydraulic head for several reasons remains constant. No-flow along a boundary naturally arise from conditions of symmetry or the abrupt truncation of geological units by very much lower permeability units. Specified-flow conditions along a boundary or at a single node are used to simulate infiltration, evaporation, or pumping of groundwater from wells.

Any of these conditions can be specified for the grid boundaries by assigning a condition to each boundary node with the help of integer maps. Node types are assigned on a small matrix representing each of the six boundaries of the flow domain. The user thus has considerable flexibility in specifying boundary conditions. The condition types are: 0 for no-flow, 1 for constant-head and 2 for specified-flow nodes. Corresponding values of flows or heads are read from the data file following the integer map for each of the six boundaries. It is also possible to use any internal node as a constant-head or specified-flow boundary with an added input list of node number and appropriate parameters.

Once the grid is assembled, individual brick-shaped elements are assigned nodal incidences as well as hydrogeologic properties including permeability, storativity, porosity and dispersivity. The incidences are ordered accordingly to a right-handed cartesian coordinate system. All these parameters can be displayed for

convenience by taking advantage of the output-option code.

Hydrogeological parameters are assigned to elements using a unit code, each unit being characterized by a parameter value. Any reasonably complex spatial structure represented by permeability or storativity values can be specified to permit the realistic description of actual physical settings. From a dimensional point of view, the user has flexibility in the use of units. Parameters are only required to be given in a consistent set of units.

Solving for Flow and Mass Transport

Once the basic set of parameters are input and generation of the mesh complete, the next operation in the program is concerned with the computation scheme for solving the groundwater-flow equation using the finite-element method. The first step of the scheme involves the evaluation of the coefficients entering the conductivity matrix. A storativity matrix is also assembled if a transient-flow simulation is run. The matrix assembly is followed by the solution to the resulting system of linear equations.

The coefficients of the matrix are obtained by numerical integration over brick elements using linear shape functions and Gaussian quadrature in all directions. The integration is done accordingly to a Galerkin formulation. This formulation is based on a principle of weighted residual in which the residual measures the innacuracy of an approximate or trial solution. If (1) is the governing

equation the residual R becomes:

$$R = F(\bar{h}) = \frac{\partial}{\partial x_i}\left[K_{ij}\frac{\partial \bar{h}}{\partial x_j}\right] - S_s\frac{\partial \bar{h}}{\partial t} = 0 \qquad (2)$$

where: F = function to be approximated;

R = residual;

$\bar{h}$ = trial solution, (L).

The essence of Galerkin criteria lies in the fact that when the weighted residual is forced to vanish nodal heads are obtained as the solution to a system of algebraic equations (Wang and Anderson, 1982). The procedure involves at first the definition of a trial solution such as:

$$\bar{h}(x,y,z) = \sum_{j=1}^{nod} N_j(x,y,z)h_j(t) \qquad (3)$$

where: h_j = head at node j, (L);

nod = total number of nodes;

N_j = weighting functions;

j = node number (1,2, ... ,nod).

The trial solution anywhere in the domain is expressed as a linear combination of surrounding nodal values. The next step is to formulate a set of conditions sufficient to find each nodal value h. These conditions are given for each node 1 as:

$$\iiint_D (F(\bar{h}))\cdot N_j \, dxdydz = 0 \qquad (4)$$

where: D = domain of integration (flow domain);

N_j = nodal basis function.

With Galerkin's method the basis functions are used as the weighting functions and (4) expresses the requirement that the weighted integrals of the residuals be zero (Pinder and Gray; 1977). Integration of (4) by parts, and the piecewise evaluation of the resulting expression on an element by element basis permits the evaluation of the coefficients entering the system of linear equations. The system has the form:

$$G\{h\} + P\{dh/dt\} = \{f\} \qquad (5)$$

where: G = conductivity matrix;

{h} = vector of unknown nodal head values;

P = storativity matrix;

{dh/dt}= vector of time derivatives;

{f} = flux or boundary conditions vector.

The coefficients are first assembled into elemental matrices and then loaded into the global matrices after a reordering of the incidences. The entries of the element matrices GE ad PE are respectively of the form:

$$GE_{i,j} = \int_{-a}^{a}\int_{-b}^{b}\int_{-c}^{c}\left[K_x \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + K_y \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + K_z \frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right] dxdydz \qquad (6)$$

and

$$PE_{i,j} = S_s \int_{-a}^{a}\int_{-b}^{b}\int_{-c}^{c} N_i N_j \, dxdy \qquad \text{with } i,j = 1,8 \qquad (7)$$

where: i = nodal contribution;

j = incidence order;

a,b,c = element half-sizes.

Entries in the flux vector have two components. The first is expressed directly in term of specified fluxes at nodes and is stored once specified fluxes are read from input. The second component results from the handling of specified-head nodes. When the head at a given node is constant the equation solving for that node is not necessary. The corresponding row j in the G and P matrices is not considered and the system of equations is condensed. The non-zero coefficients of row j in G are then loaded at the appropriate locations in the flux vector.

The evaluation of the integrals is most conveniently done in a local (ξ,η,ζ) system of coordinates. A transformation from local to global coordinates requires that the derivatives be transformed according to a chain rule. The inverse Jacobian matrix is used for this purpose and the determinant of the matrix is used to scale down the elemental size. Gaussian quadrature is used to compute the integral, which is expressed as a summation over two points for each coordinate direction.

Details on the integration and elemental assembly procedure are treated in Wang and Anderson (1982) for two-dimensional applications. The numerical scheme in 3DFT is similar to the one used in another three-dimensional finite-element code, ISOBRIQUE, developed by Verge (1975) for isoparametric elements. However a more complicated treatment of incidences permits ISOBRIQUE to deal with

higher and mixed-order elements. 3DFT uses only the simpler linear elements. The major difference between ISOBRIQUE and 3DFT is in the assembly and storage of the conductivity (G) and storativity (P) matrices.

Both codes make use of the symmetric and banded nature of G and P matrices in order to reduce the storage to the upper half-bandwidth. However ISOBRIQUE uses a 'full-storage' solver for banded and symmetrical matrices, and 3DFT uses the important sparsity of finite-element matrices to further compress the storage by retaining strictly non-zero coefficients. The coefficients are loaded in vectors instead of matrices and two additional vectors are needed to keep track of their original row and column location in the banded matrices. The use of a sparse-matrix solver permits substantial savings in storage and computing effort.

The time discretization scheme is of the form shown in (8). Because of the way it is implemented in the code it is possible to choose the kind of discretization by specifying the value of TD in:

$$\left[G + \frac{TD}{\Delta t} \cdot P\right] \{h\}_{t+\frac{\Delta t}{TD}} = \frac{TD}{\Delta t} \cdot P \{h\}_{t} + \{f\} \qquad (8)$$

$$A \{h\}_{t+\frac{\Delta t}{TD}} = \{f\} \qquad (9)$$

where: TD = time discretization variable (1 or 2).

The fully implicit procedure is used when TD is one whereas TD equal to two specifies a centered-difference

interpolation. The first member of the left hand side of (8) is stored in one array A and the right hand side is stored in a vector {f} so that a system of the form (9)) results. The solution to this system requires that A (i.e. G and P) be assembled only once since equal time increments are used. Only the {f} vector needs to be updated and reassembled at each time step. The software package providing the solution to the linear system, by Eisenstat et al. (1982), uses a Cholesky decomposition of the coefficient matrix in an upper and triangular matrices followed by a backsubstitution. It operates on compressed storage and yields the solution in a vector form in {h}.

For a steady-state run the matrix P is not assembled and the simpler system is solved directly when provided with the proper boundary conditions. Any solution to a transient-flow problem requires that initial heads be specified at each node. These values can be specified as input for simple problems or, for more complex problems, can be obtained by solving the system's steady state and by using the results as initial values. The result of these computation steps can all be printed for verification.

Once the distribution of hydraulic head is obtained, the transport simulation can take place. Reference particles are inserted in the flow field by specifying rectangular areas along any boundary of the flow domain. On every surface that acts as a source, the particles are located randomly at the start of every transport time-step. In this

respect, 3DFT is similar to the code TRANS developed by Prickett et al. (1981). The actual size of these source areas can vary from the entire side of a given domain to a single line or even a point.

Contaminant loading-function can be generated to replicate complex cases by adjusting the size of time-steps between each particle loading and by varying the number of particles generated at each step. In this work however, mostly continuous sources are utilized.

Particles are moved through the flow field one at a time during a given time-step. First, the advection is accounted for by a deterministic motion calculated from the components of the velocity vector at the particle location and from the magnitude of the time-step. Then the dispersion is taken in account by adding a random component to the deterministic motion.

There are three components to the velocity vector, one for each coordinate direction. Each of these components is evaluated at the particle location (1) by defining its value between node pairs surrounding the particle using Darcy law, (2) by attributing these values to points lying midway between the grid nodes and serving as a local reference points and (3) by interpolating linearly between these values as a function of the particle location with respect to those reference points.

For each velocity component the interpolation is made from 8 surrounding internodal reference points: 4 points

from the element containing the particle and 4 others from the closest element in the same coordinate direction. Evaluating velocities in this way results in a smoother and more realistic variation in values between units of contrasting permeability. When a particle is close to one of the domain boundaries there is just one element available for interpolation. In such a case fictitious interpolation velocities are generated. Fictitious velocities are chosen in order to insure that flow remains parallel to no-flow boundaries and converges toward flow nodes. They also permit particles to leave or to enter the domain through constant-head boundaries.

Once the velocity vector is known the particle is displaced in a deterministic way as a function of the vector magnitude and displacement time. The extent of any single displacement is controlled by the user who fixes the duration of the steps of the transport simulation and who defines the minimal number of such displacement required for a particle to travel within individual elements. This feature enables particles to be displaced more accurately on curved flow paths because small linear increments are used. It also prevents a ricochet effect where particle are advected far beyond the domain boundaries when a poor estimate of time-step size might cause moving particles to move across more than one element.

The effects of dispersion are accounted for by adding a random component to the deterministic motion of particles.

The relative size of this random motion is related to the dispersive properties of the porous medium (Ahlstrom et al., 1977). Reference particles are relocated from their deterministic position by calculating the displacements in all three coordinate directions using the following equations (Ahlstrom et al., 1977):

$$XR = XD+(D_1 \cdot VX/V)+(D_2 \cdot VY/V)+(D_3 \cdot V2/V) \quad (10a)$$

$$YR = YD+(D_2 \cdot VY/V)+(D_1 \cdot VY/V)+(D_3 \cdot V2/V) \quad (10b)$$

$$ZR = ZD+(D_2 \cdot VZ/V)+(D_3 \cdot VY/V)+(D_1 \cdot V2/V) \quad (10c)$$

The parameters are function of the longitudinal and transverse dispersion coefficients:

$$D_1 = \sqrt{24 \cdot \xi_1 \cdot \Delta x}\ (.5-[Z]_0^1) \quad (11a)$$

$$D_2 = \sqrt{24 \cdot \xi_2 \cdot \Delta x}\ (.5-[Z]_0^1) \quad (11b)$$

$$D_3 = \sqrt{24 \cdot \xi_3 \cdot \Delta x}\ (.5-[Z]_0^1) \quad (11a)$$

where: XR,YR,ZR = particle coordinate after random motion;

XD,YD,ZD = particle coordinate after deterministic motion;

VX,VY,VZ = velocity components (L/T);

V = velocity magnitude (L/T);

D_1, D_2, D_3 = random displacement parameters;

ξ_1 = longitudinal dispersion coefficient (L);

ξ_2, ξ_3 = transverse dispersion coefficients (L);

Δx = total displacement (L);

$[Z]_0^1$ = random number between 0 and 1.

Equations (10) and (11) are derived in Ahlstrom et al. (1977) and the reader is referred to this work for more

detail. This formulation of the dispersion process is valid for porous media which are isotropic with respect to hydraulic conductivity. The flow systems handled by 3DFT can be anisotropic, but because of the lack of practical dispersion model generally applicable to these systems, the use of the isotropic formulation is extended here as a first approximation to the anisotropic case.

Associated with the particle-tracking techniques are problems in handling the probabilistic motion of particles in a system with interlayered units of contrasting hydraulic conductivity, which produces marked contrast in velocity. The problem typically arises with lateral dispersion when flow is parallel to the interface between the units. In such a case the extent of the lateral motion is controlled by the magnitude of the velocity which can be orders of magnitude greater in the high-conductivity unit. Large lateral motions can cause some particles to be relocated far into low-velocity units. The magnitude of subsequent motion is then usually not sufficient to permit a comparable motion back in high-velocity zones. Thus, an improper handling of the lateral component of dispersion can cause an artificial accumulation of mass marginally to the preferential flow zones.

Features developed in 3DFT alleviate this problem by adjusting the lateral components of the random motion according to the actual velocity contrast between units. By moving particles in this manner no particles can get

artificially isolated from the main flow system because the lateral motion is reduced by proportion to the ambient velocity contrast on each side of the interface. Another factor that helps preventing exagerated particle motion is the limitation imposed on the advective motion which forces individual dispersive steps to be smaller.

The particle motion must also account for the boundary conditions for the mass-transport problem. When lateral dispersion carries a particle out of the domain boundaries along a no-flow boundary, the particle is simply reflected so as to preserve the randomness of the distribution. When the longitudinal dispersion causes a particle to leave the domain, the particle is relocated at the domain boundary so as to preserve mass balance. Futher transport will continue to occurs until (1) a constant-head boundary, (2) a pumping node or (3) a sink is reached.

Within the domain, particles may be transported to a sink defined by the user. Particles are actually removed from a volume surrounding a labeled node. The volume of an individual sink is the same dimensionally and geometrically as a grid element, but a labeled node occupies its center. Particle removal helps to avoid unrealistic mass accumulations. Removing particles also reduces storage requirements and computing effort that would have been required to handle particles having terminated their advective displacement near a discharge boundary. The procedure used for particle removal is identical to the one

used by Prickett et al. (1981).

The number of particles removed from the transport simulation at any sink can be printed at the end of each time step. Specification of sinks is entirely problem dependent and is thus left to the user. Labeled nodes can be specified as separate sinks or can be grouped. For example, all the nodes at a boundary of the domain can be assigned the same sink number. With such a configuration, the arrival time of a whole particle swarm can be displayed. The only input required is the node number of each labeled node together with the specification of the sink it represents.

The particle distribution is stored after each transport step and can be printed separately together with a numerical map of mass concentration. Concentration is computed from the total mass of particles in each element divided by its pore volume. The mass of an individual particle represents the smallest amount of contaminant that can be tracked and hence limits the amount of detail that can be obtained from the contamination pattern. If the flow through the source is known, as such is the case for uniform flow systems, particle mass can be adjusted to the contaminant-loading function so as to simulate a predetermined source concentration.

During each time step a given pore volume of water is being flushed through the source. This volume has to receive the contaminant load needed to bring the concentration to the predetermined value. This load has then to be subdivided

between the number of particles required for an appropriate resolution of the plume.

III. EVALUATION OF THE MODEL

The purpose of this chapter is to describe the steps taken to evaluate some aspects of the operational efficiency of the model and to verify the accuracy of the computed results. The first section is a survey of 3DFT storage and effort requirements, the second section presents an account of flow simulation tests and the third section analyses the results of transport simulations.

A. Storage Requirements and Computer Effort

The storage needed for the program itself amounts to 75 kilobytes. This number is the minimum memory size for storing the compiled execution commands in a machine-language form. The actual memory required for a real problem depends mainly on the size of the finite-element grid because hydraulic parameters must be assigned to each element of the grid, and a workspace must be available to solve the finite-element equations. In addition, the particle-tracking procedure, explained in Chapter Two, also requires that information be stored for each particle. For these reasons, it is obvious that memory size will be extremely variable from problem to problem.

As an example, examine Figure III.1 which shows the storage needed for the execution of a 500 node flow simulation and the transport of 5000 reference particles. For illustration purposes the computer code's 22 subprograms are regrouped in 6 essential segments each dealing with one

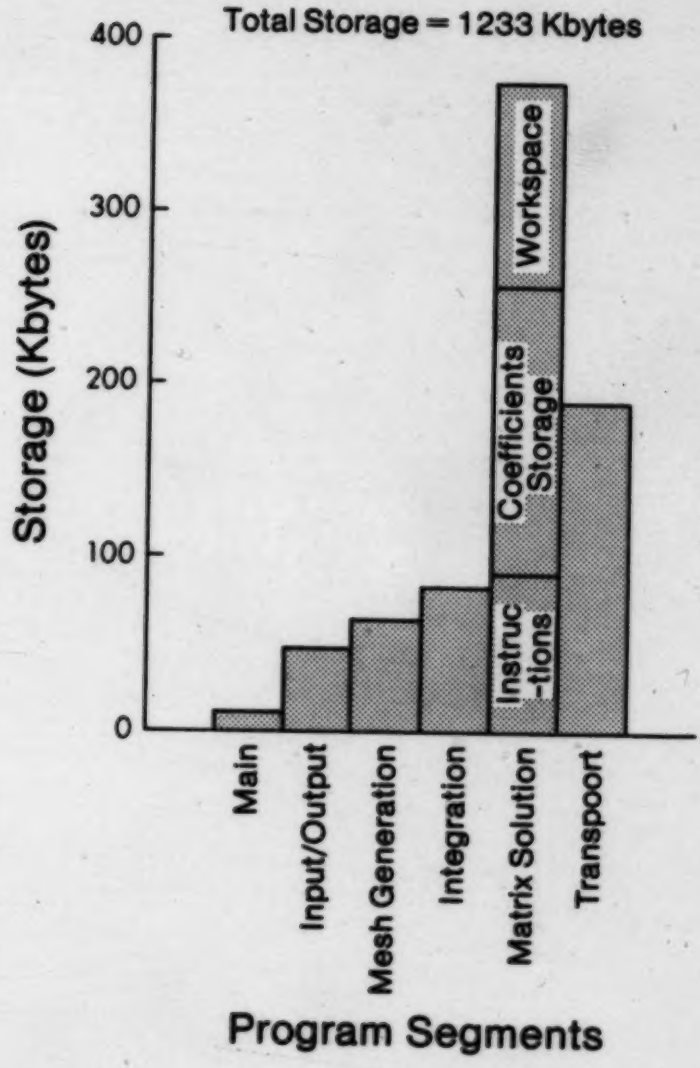


Figure III.1
Illustration of the core requirement (in kilobytes) for various program segments in a 500 node steady-state flow simulation.

of the phases of the program execution outlined in Chapter Two. The total storage in this case is 1233 kilobytes. The most important part of this memory requirement is linked to the matrix solver which handles the system of linear equations. Most of that memory is necessary to store the matrix coefficients and to provide a workspace for the solver.

The storage required for the coefficients is determined by the number of equations to be solved (one per unknown head). The extreme structural sparseness of the finite-element matrices allows for substantial reduction in memory. Hence, the storage of coefficients is compressed at the moment of assembly and this, in turn, permits the size of coefficient arrays to be reduced to a reasonable minimum.

The possibilities for storage economy related to coefficient storage are typically problem-dependent. For a small finite-element grid and few unknown heads the saving is negligible but it increases rapidly for larger grids. In fact, it then becomes proportional to the number of equations (NF).

The second most important storage requirement is related to the workspace, an array needed for the direct solution to the linear system of equations. The array dimension NSP is defined by :

$$NSP \geq 0.1\,(NB \cdot NF^{1.4}) \qquad (12a)$$

and

$$NSP \geq 50 \cdot NF \qquad (12b)$$

where: NSP = workspace dimension;

NB = matrix bandwidth;

NF = number of equations.

The combined effect of using a compressed storage mode for the coefficients and using a workspace is that the benefits of compression are compensated for by the workspace requirements. The remaining storage for this segment is taken by the matrix solver itself, which is not large and is insensitive to the problem size.

Another important program segment is the transport code. It may also require large amounts of storage. Figure III.1 illustrates the storage requirements for tracking 5000 particles combined with 50 sinks. In a case where much larger numbers of particles are required, which can happen for three-dimensional simulations, most of the total storage could be affected to this segment. This storage cannot be compressed.

The main advantage of a compressed-storage mode combined with a sparse-matrix solution is in the cost-effectiveness of the simulation. A considerable improvement over dense-storage mode is found in relation to the elimination of useless operations on zero coefficients. The efficiency gain is proportional to the size of the problem to be solved. As an example, Table III.1 shows a comparison of the execution cost in dollars between sparse-solver and dense-solver versions of the code for two

SOLVER	PROBLEM SIZE (NUMBER OF NODES)	
	(60)	(500)
Dense	.16	5.12
Sparse	.20	3.31

Table III.1 Comparison of the efficiency between the dense and the sparse solver. Numbers show the execution cost in dollars.

problem sizes. These results indicate that for larger problems the marginal computer effort needed to keep track of the coefficients in a compressed storage-mode is more than compensated by the overall saving in execution effort.

For a typical flow simulation the relative effort is not equally distributed between each segment. As illustrated in Figure III.2, the most important effort is generally devoted to the integration of the matrix coefficients and to the solution of the system of linear equations. Various input and output operations also generate moderate costs. For a steady-state simulation (Figure III.2B,) the effort is nearly equally divided between the integration and the solution segments. For the transient-state simulations their relative importance depends on the number of time steps because the integration is performed only once while the solution to the head distribution is repeated for each step.

As shown for a transient solution using one time step, the integration effort is relatively more important than for the steady-state case because the coefficients of the storativity matrix, equation (7) must be computed in addition to the conductivity matrix, equation (6). The solution effort at each step is not related to the type of simulation because the system solved remains of the same form (see equation (9)) for both steady and time-varying systems. The relative importance of the solution effort however does increase with the number of time steps. For transient simulations the relative effort of input-output

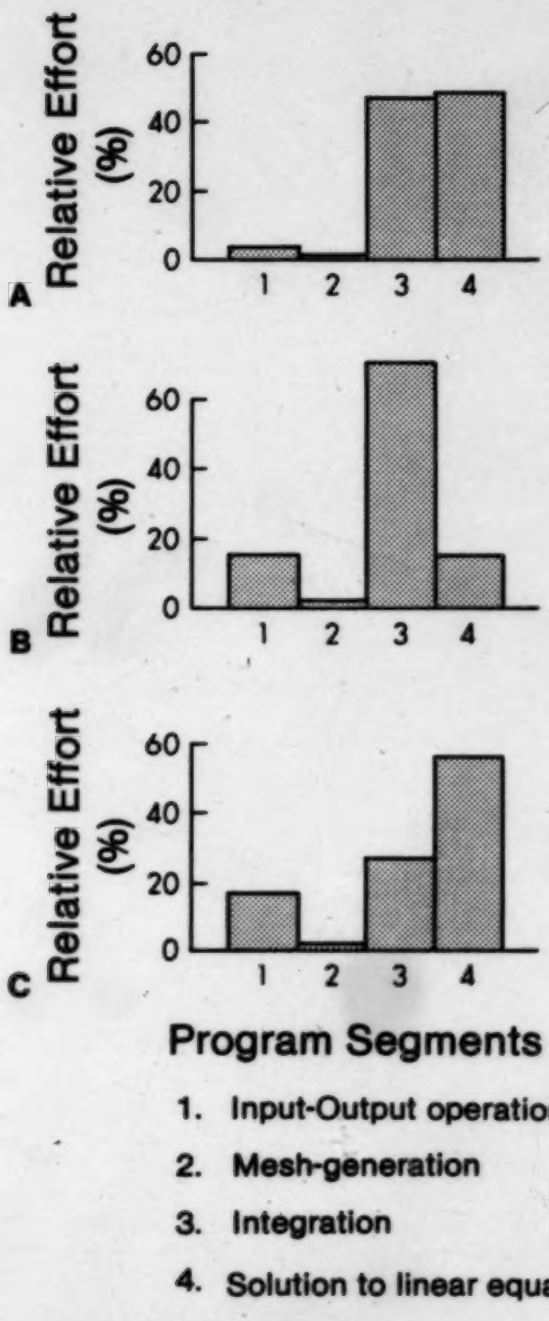


Figure III.2
A comparison of the relative computer effort for each segment of the code for typical steady state (A), and transient-state simulations with one time step (B) and 10 time steps (C).

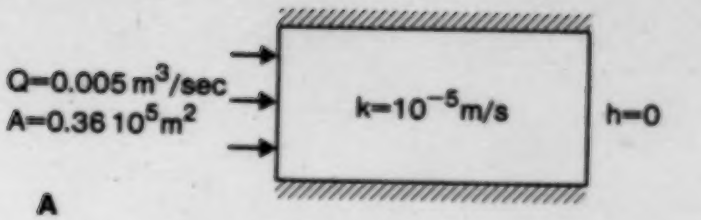
operations does not seem to be affected by the total number of steps because of the proportional number of read and write operations then involved.

The execution of the most demanding subroutines can be transferred to a peripheral array processor with very minor modifications to the original code. Indications are that the economy is important and can reach 400 to 500% with respect to the execution entirely on the main-frame computer. This result suggests that for the execution of large problems the user can eventually rely on this device for increased cost efficiency.

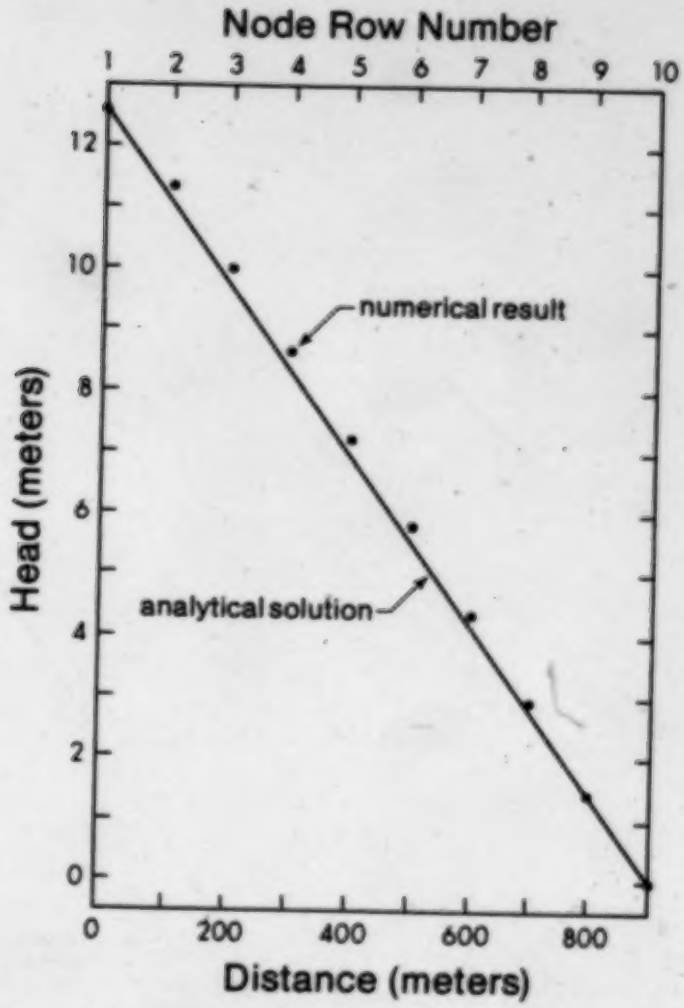
B. Verification of the Flow Simulation Results

Steady-State Flow

The accuracy of 3DFT was evaluated for various boundary and initial conditions by comparing model results to known analytical solutions for both steady and transient-flow simulations. The first test problem consists of a uniform-flow domain created using a constant-head boundary at one end and by a recharge boundary at the opposite end (Figure III.3A). The other sides are no-flow boundaries. The flow conditions are uniform along the domain. The grid consists of 10 X 5 X 5 nodes with the head drop occuring along the 10-node axis. For this problem, the solution is obviously a linear head decline between the recharge and constant-head boundary. The computed values are in close



A


Node Row Number
1 2 3 4 5 6 7 8 9 10
Head (meters)
0 2 4 6 8 10 12
numerical result
analytical solution
0 200 400 600 800
Distance (meters)

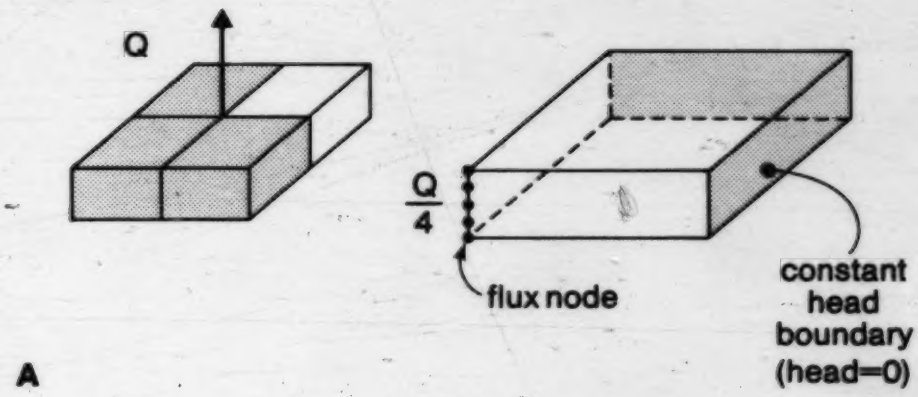
B Head distribution along the flow domain.

Figure III.3
Figure showing the setup and results of a simulation of flow in a simple rectangular domain.
A Illustrates the boundary conditions (Q = flow across the crossection A, h constant head, k permeability).
B Shows the finite-element results compared to the analytical solution for the same problem.

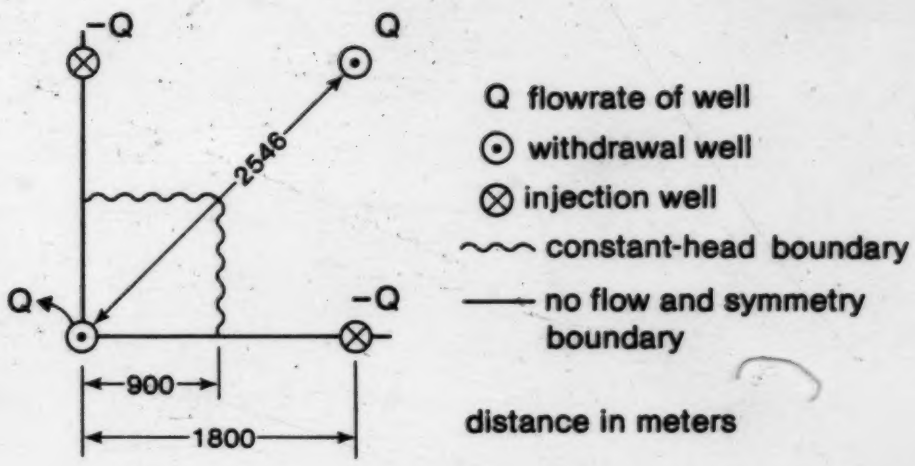
agreement with the expected ones. The maximum deviation is 2% of the total head drop (Figure III.3B).

Another important test with 3DFT is the "island discharge" problem. Figure III.4A illustrates a well situated at the center of a square island bounded by constant-head boundaries. Properties of symmetry permit us to consider only one quarter of the flow domain in the analysis. Reducing the size of the domain to one quarter of the original size allows a considerable storage reduction and overall computing economy. The setup shown in Figure III.4A uses 5 flux nodes at one edge of the 10 X 10 X 5 grid. Opposed to the flux nodes are two vertical constant-head boundaries and the remaining sides are no-flow boundaries. One fourth of the total well discharge Q is distributed among the 5 flux nodes.

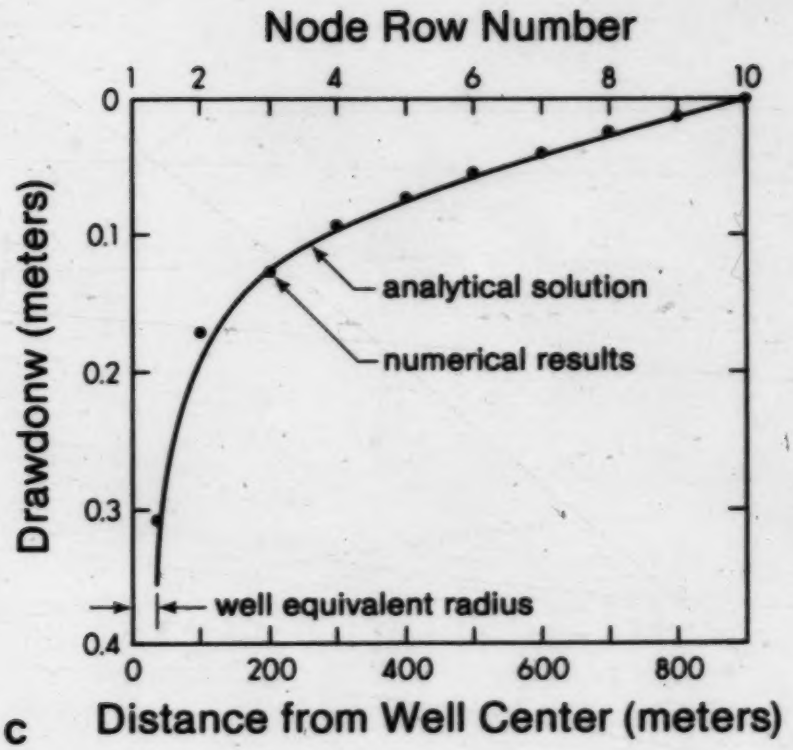
The exact solution to this problem is given by image-well theory by using the construction shown in Figure III.4B. Recharge boundaries act as imaginary wells symmetrically opposed to the discharge well but of reversed yield. Two such recharging image-wells are shown in Figure III.4B. Boundaries are also mutually reflected and second-order image-wells can be generated from the first two and so on. By virtue of the superposition principle the resulting drawdown at any point in the domain is the combined effect of all the real and image wells. However because the wells have limited influence beyond which no perceptible drawdown can be measured, the contribution of


Q
Q/4
flux node
constant head boundary (head=0)

A


-Q
Q
2546
Q
-Q
900
1800
Q flowrate of well
withdrawal well
injection well
constant-head boundary
no flow and symmetry boundary
distance in meters

B



C

Figure III.4
Illustration of the "island-discharge" problem.
A A conceptual square island and the equivalent setup using properties of symmetry.
B The image-well reconstruction used to formulate the steady-state solution.
C A comparison of theoretical and numerical solutions using a 10 X 10 X 5 grid.

distant image-wells quickly becomes negligible. The radius of influence can be estimated using Sichardt empirical relation:

$$R = 3000\ s\ (k)^{0.5} \qquad (13)$$

where: R = well radius of influence, (meters);

s = drawdown, (meters);

k = conductivity, (meters/sec).

As a first approximation R can be computed by using the drawdown at the well as computed by 3DFT (0.358 meters). Because k=1 the radius of influence is estimated at 1059 meters. This value indicates that the pumping well is too far from the recharge boundaries for the drawdown to be affected. Consequently, it indicates that the first approximation is accurate. Thus the calculation of the analytical result can be performed using the image-well formula for one well:

$$s = \frac{Q}{2\pi km} \ln \frac{R}{r} \qquad (14)$$

where: Q = well discharge, (meter^3/sec);

r = effective well radius, (meters);

m = aquifer thickness, (meters).

For calculation purposes an effective well radius of 20.8 meters is used. It corresponds to a 'point' discharge as simulated using a square grid with 100 meters spacing between nodes (Rushton and Redshaw, 1979).

Comparison between the theoretical drawdown and 3DFT numerical values taken along the symmetry boundaries is illustrated in Figure III.4C. The agreement is very good with a maximum error of about 1% except near the well where high gradients and highly converging flow makes the numerical solution more difficult and causes an error of up to 15%. The poor performance of numerical techniques in the vicinity of the pumping is typical of virtually all numerical approaches.

Transient-State Flow

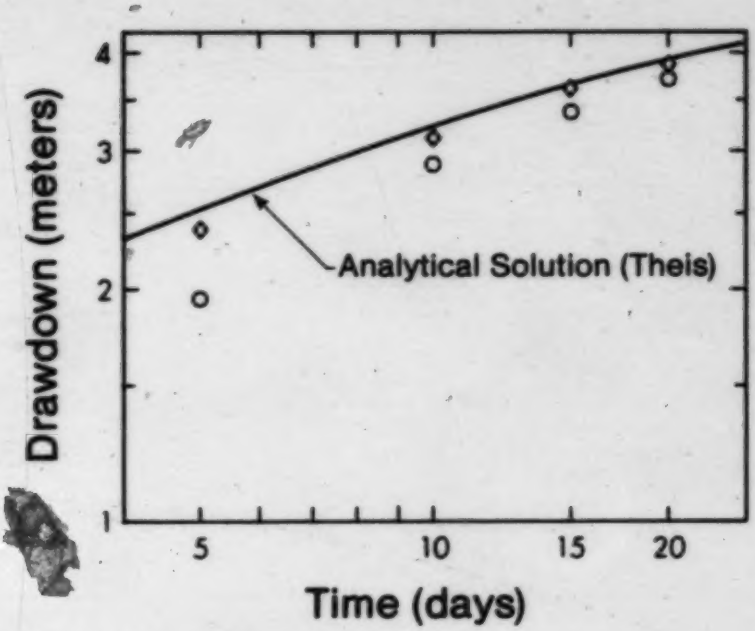
The capabilities of 3DFT to simulate transient-flow conditions are evaluated by comparison to the equation for transient and artesian aquifer as provided by Theis (1935). This solution applies to an isotropic aquifer of uniform thickness and infinite extent. The grid used is analogical to the steady-state discharge problem (Figure III.4) but it is comprised of 17 X 17 X 2 nodes, horizontal spacings of 1000 meters in each direction, two corner flux nodes and no-flow boundaries. By specifying these boundary conditions, the solution is only really comparable to the analytical solution for those times before the cone of depression reaches the boundaries opposite to the pumping well nodes.

The numerical results are compared with Theis solution for a time period of 20 days. Over this time, drawdown was observed to spread from the two flux nodes towards the opposite end of the grid. Results for radial distances of

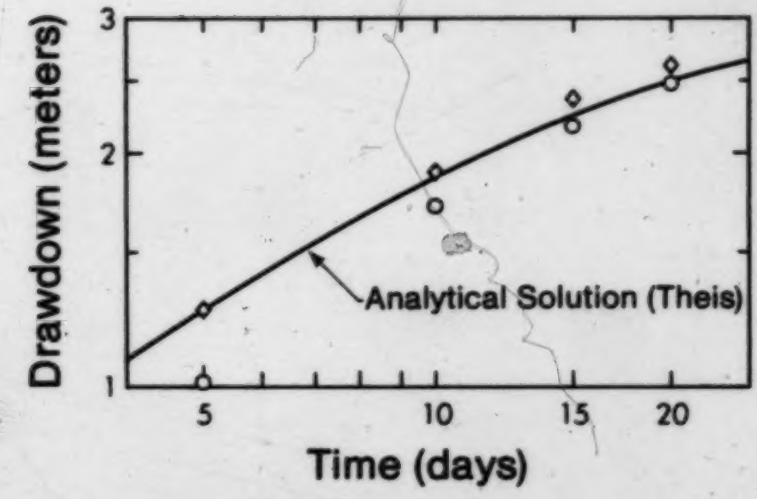
1000, 2000 and 10000 meters are shown in Figure III.5. The most common observation is that a minimum number of time steps (usually less than 5) is required for the numerical solution to converge to within one or two percent of the analytical solution. A similar warm-up factor is commonly observed with finite-difference approximations such as Prickett and Lonnquist (1971) flow-simulation code as well as with other finite-element codes such as Verge's three-dimensional model (1975).

Also apparent in the results is that the accuracy of the solution changes as a function of the distance from the well. The best results are obtained for observation points at greater distance from the well (Figure III.5C). Here, changes in hydraulic head are smaller as are hydraulic gradients.

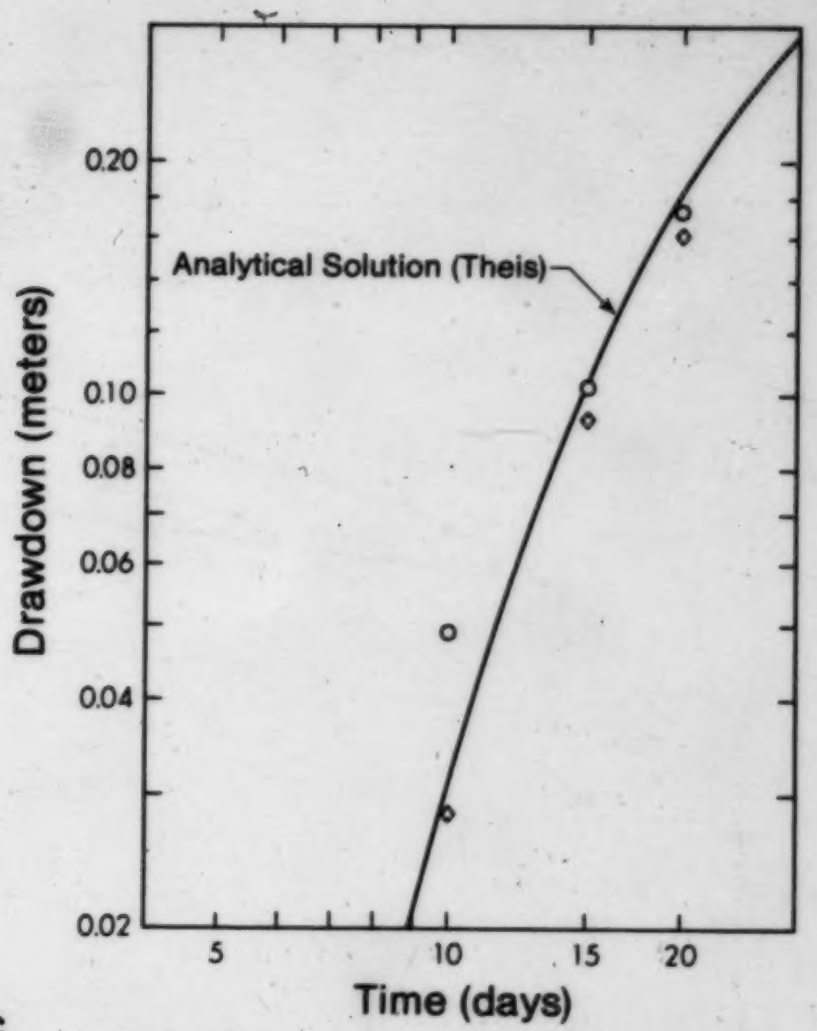
The sensitivity of the convergence rate to the time step size is illustrated by comparing the solutions obtained using a time step of 5.0 days and 0.5 days (compare computed drawdowns in Figure III.5). When using a time step of 5.0 days, four steps are generally required to reach convergence within a few percent. The use of a smaller time step, such as 0.5 days, accelerates the convergence. But although convergence is faster, the relatively larger number of steps used causes an accumulation of numerical dispersion that increases the error after convergence is reached. Such numerical dispersion tends to be more important far from the well where smaller head change occurs, as is apparent in



A r = 1000 meters



B r=2000 meters



C

r=10,000 meters

◇ computed drawdown using DELT = 0.5 days
○ computed drawdown using DELT = 5 days

$Q=1.337\ 10^5\ m^3/day$

$T=10^4 m^2/day$

K=10 m/day

$S=10^{-2}$

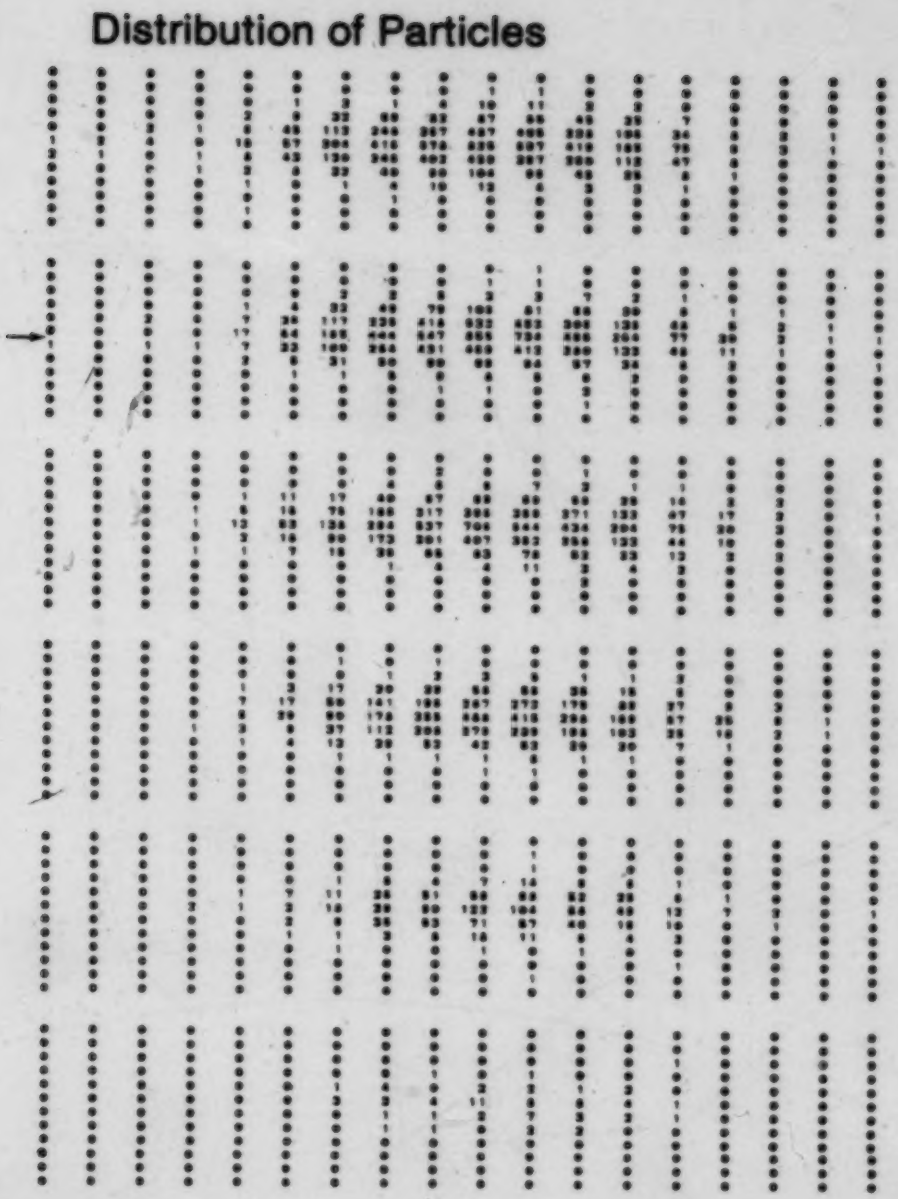
Figure III.5
Time-drawdown comparison of theoretical and digital computer solutions using two different time step intervals DELT. Time-drawdown results are shown for a distance r=1000 (**A**), 2000 (**B**) and 10000 meters (**C**).

Figure III.5B and C. Therefore, the appropriate time step appears to be a trade-off between convergence rate and numerical dispersion, the final choice depending solely on the time at which the accuracy of the solution becomes critical.

C. Verification of the Mass-Transport Simulations

Particle location within the flow domain can be output after each transport step in the form of a numerical map showing the total number of particles that are in each element. A typical output is shown in Figure III.6 for a grid made out of 6 layers of 19 by 13 elements. A uniform velocity field of 0.274 10^{-3} cm/sec is simulated, so that the flow is directed from the source side (left) to the opposite end of the grid (right). The uniform flow field is generated in this case by assigning constant head boundaries to each end of the rectangular domain.

In the case illustrated, a pulse of 32000 particles is injected from a point source indicated by an arrow. When such a large number of particles is used the distribution, after some time, tends to be Gaussian in each direction. The center of mass is the point of maximum particle density. The swarm of reference particles migrates with the average seepage velocity at the geometrical center of an elliptical-shape plume. As the particles leave the source and are advected downflow (from left to right) they tend to spread away from the center of mass and in every directions



Velocity =0.000274 cm/sec, time = 16.8 years (5.32 10^8 sec),
ξ_x=260 cm, ξ_y=220 cm, ξ_z=7.3 cm, → source location

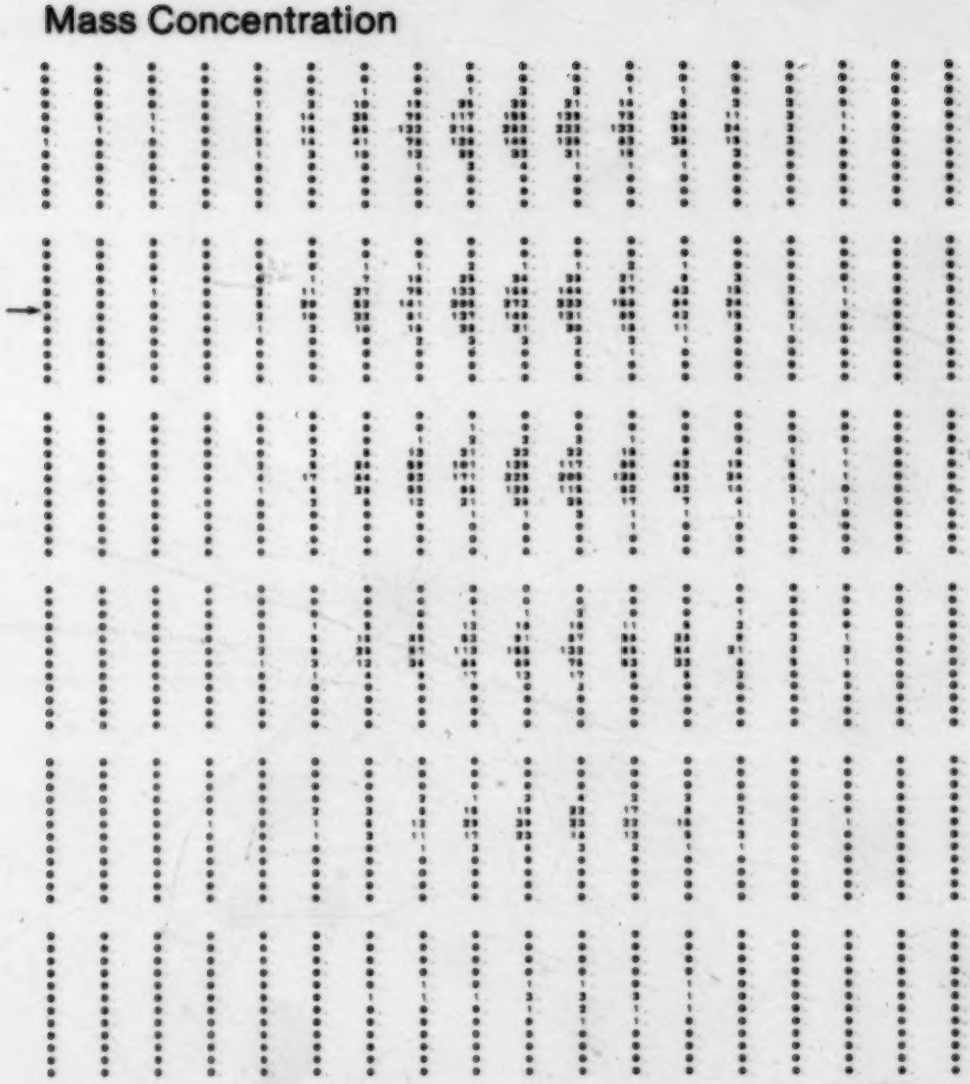
Figure III.6
The particle distribution in each element of the grid resulting from the instantaneous injection of 32000 particles at a point on the left end of the grid. The layers of the grid are each comprised of 12 rows and 18 columns of elements.

as a function of velocity and dispersivity coefficients. In this case, because the source is close to the upper boundary of the flow system, the plume is partially confined between the top and bottom no-flow boundaries. Part of the spread is thus forced to occur downward as some particles are "reflected" during their upward dispersive motions.

Figure III.6 shows the location of the center of mass after 27 years. It is located at the 7th row and 10th column of the 5th layer from the bottom. Concentration is calculated from the particle distribution on the basis of particle mass and element pore volume. Figure III.7 illustrates the extent of contamination corresponding to the particle distribution of Figure III.6. This numerical output is a contourable map of concentration in parts per million or milligrams per liter.

One-dimensional Simulation

Just as the flow code is tested using analytical solutions so to is the mass transport algorithm. The one-dimensional verification has been carried out by adding particles to a uniform velocity field considering the effect of advection and longitudinal dispersion only. Lateral dispersion in the y and z-directions is eliminated by setting the corresponding dispersivities to zero. For this problem the analytical solution to time-varying concentration C along the flow axis x, after Ogata and Banks (1961) is:



Element volume=1.888 10^{11}cm^3, particle mass= 6.0042 10^{10}mg,
unit porosity, → source location

Figure III.7
Numerical map of mass concentration 16.8 years after the instantaneous input of mass at a point source. A particle mass of 6.0042 10^{10}mg is used. Each particle in effect represents a concentration of 0.32 mg/ℓ.

$$C(x,t) = 0.5\, C_o\{\mathrm{erfc}[(x-vt)/2(Dt)^{0.5}]\} \qquad (15)$$

where: C = concentration at point x and time t,(M/L^3);

C_o = source concentration, (M/L^3);

x = distance, (L);

t = time, (T);

v = seepage velocity, (L/T);

D = coefficient of longitudinal dispersion (L^2/T);

erfc = complementary error function.

The coefficient of dispersion in this case is the product of the longitudinal dispersivity and the seepage velocity.

The boundary conditions for the problem are shown in Figure III.8A. The concentration at the source remains constant at C_o throughout the simulation but is zero everywhere else at time equal zero. The continuous source is simulated using short time steps with a small number of particles injected at each step. Results of the simulation are shown in Figure III.8B after 487.3, 974.6, 1461.9 and 1949.2 days. Fifty time steps are elapsed and 1500 particles are injected between these periods. The mass of each particle is adjusted so that the concentration remains constant and equal to one at the source given the pore volume flow at each time step.

Both analytical and numerical results show the progression of a concentration front at the same velocity as the groundwater. The progressive smearing due to dispersion is reflected in both cases by the decreasing slope of the

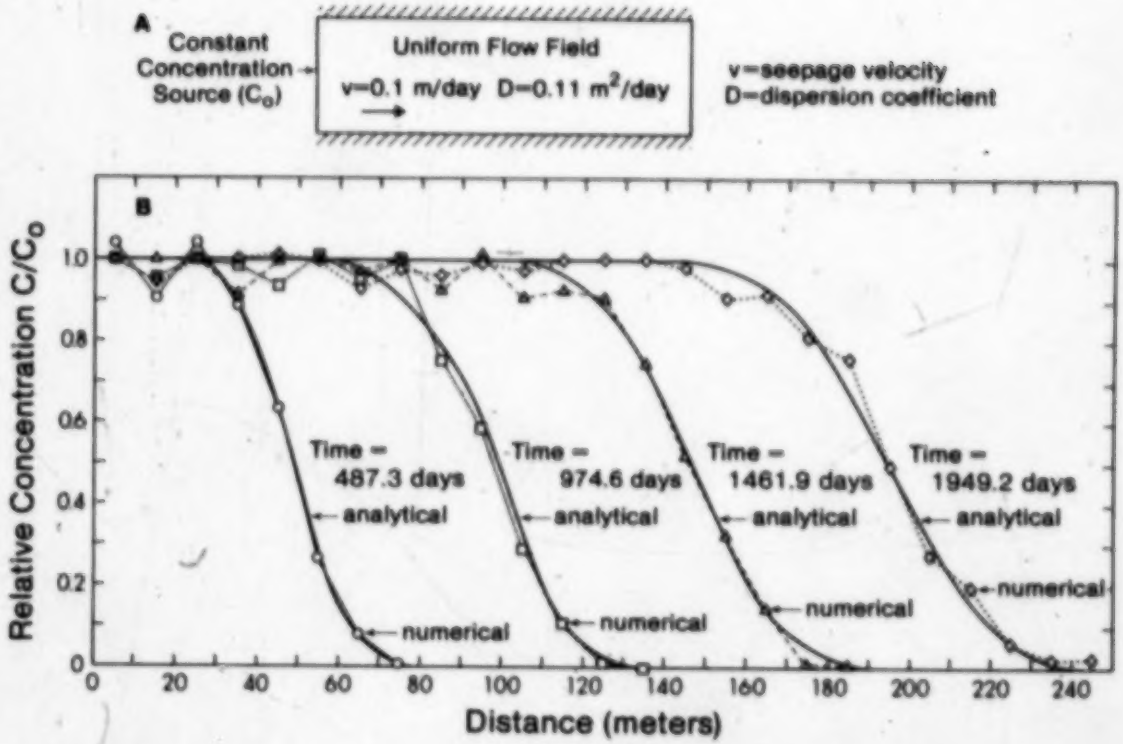


Figure III.8

The results from 3DFT compared with the analytical solution to the one-dimensional advection diffusion equation using a time step of 9.74 days and 30 particles per step, diffusion coefficient D=0.11 m^2/day.

A The boundary and initial conditions of the system: the concentration at the source is constant (C_o) with the initial condition C=0 at time =0.

B Shows the relative concentration (C/C_o) along the flow axis at various times as obtained from the analytical solution (solid line) and the 3DFT model (broken lines).

front. The front is followed by a maximum-concentration zone representing the steady state.

The accuracy of the numerical solution is in part a function of how well the particles are mixed in each element. In general, there is not a good mixing at the source. As time goes on, better mixing occurs and, generally, a more even particle distribution is obtained within individual elements. The progressive improvement of the mixing can be illustrated by the standard deviation around the mean for the steady-state part of the curve, which progressively decreases from 8% to 4% between time of 487.3 days and 1949 days. For the same period, the standard deviation of the overall results including the concentration front, remains within 3% of the analytical solution. These results are very close to those obtained with a comparable two-dimensional code (Schwartz and Crowe, 1980).

Three-dimensional Simulations

In order to fully verify the code it is essential to carry out verification trials in which a three-dimensional plume is developed. Among the analytical solutions available for these tests one of the most interesting deals with continuous aeral sources releasing contaminant in a constant and uniform flow field. Such situations are found under a variety of conditions of practical interest for example at landfill sites where the leaching of buried wastes continually introduces a mass of dissolved constituents. The

three-dimensional trials in this study are based generally on a set of data for a landfill located on Long Island, New York. This site was studied by Strautman (1984). This latter work involved calibration of analytical models against observed concentration patterns and their use for determining dispersivity and contaminant-inflow parameters. Thus for this plume both analytical and numerical models are available. Here, the analytical description of the plume is of interest.

Because the three-dimensional simulation makes reference to the Babylon site, some of the parameters are expressed in feet to correspond to the original data. The analytical solution we use is a modified form of the finite-source equation (equation(1) in Domenico and Robbins (1984b)) :

$$C(x,y,z,t) = C_o/8 \cdot \{\mathrm{erfc}[(x-vt)/2(\xi_1 \cdot t)^{0.5}]\} \qquad (16)$$

$$\{\mathrm{erf}[(y+Y/2)/2(\xi_2 \cdot x/v)^{0.5}]$$

$$-\mathrm{erf}[(y-Y/2)/2(\xi_2 \cdot x/v)^{0.5}]\}$$

$$\{\mathrm{erf}[(z+Z/2)/2(\xi_3 \cdot x/v)^{0.5}]$$

$$-\mathrm{erf}[(z-Z/2)/2(\xi_3 \cdot x/v)^{0.5}]\}$$

where: $Y/2$, $Z/2$ = half dimensions of the source area (L);
erf = error function.

This solution can be calibrated against a plume observed to spread from a constant source in a uniform-flow

field. Once the calibration is done, a relatively straightforward analysis permits to isolate the dispersivities of the porous media for longitudinal and transverse directions as well as an estimate of the total contaminant input.

Following Strautman (1984), I have utilized the following transport parameters: the longitudinal dispersivity is 260 cm, and the lateral dispersivity is 220 cm horizontally and 7.3 cm vertically. Seepage velocity is $0.274\ 10^{-3}$ cm/sec and the source function C_oQ is estimated at $9060\ 10^{3}$ mg/sec of dissolved contaminants. The product of volumetric injection rate Q and the source concentration C_o (C_oQ) is known from a detailed evaluation of the actual concentration distribution in the plume, but neither component can be isolated through analysis of the data. Flow is assumed to be uniform along the x-axis, which thus becomes the center axis of the plume. The finite-source equation (16) requires the specification of the source in term of half width (Y/2) and half height (Z/2) around the source center situated at the origin of the reference system (x=0, y=0, z=0). The flow term Q is the product of the source-area size A times the seepage velocity v.

Although any combination of C_o and Q could, in principle, produce the same contaminant input, the actual specification of source concentration and volumetric injection rate is critical for the verification of the model. The effect on numerical results caused by modifying

the size of the source area, while using the same contaminant load (i.e. number of particles and particle mass) is shown in Figure III.9.

The concentration along the plume axis is shown to be seriously affected by the source size. A much greater initial dilution occurs as the contamination is spread over an increasingly large area. The dilution is about 50% greater with the 3272 m^2 area than with a point source and the effect is felt in the entire length of the plume. An area of 3272 m^2 is chosen for simplicity because it corresponds to a source concentration of 1000 mg/l with a total C_oQ of 9060 10^3 mg/sec.

The general features of a three-dimensional plume in a uniform flow field as a result of contamination from a continuous areal source are depicted in Figure III.10. The contours are plotted from a numerical output similar to Figure III.7 but obtained using a continuous source. The source and porous media parameters used are those defined previously from the Babylon site, at the exception of the thickness of the aquifer which is increased in order to avoid any confinement of the plume between the upper and lower boundaries. Figure III.10 shows the extent of contamination in the flow field after 6.75, 13.5 and 27 years. The concentration patterns are outlined for a horizontal plane within the aquifer situated at the level of the center axis or at elevation zero (A, B and C). The spread in the vertical direction results in similar

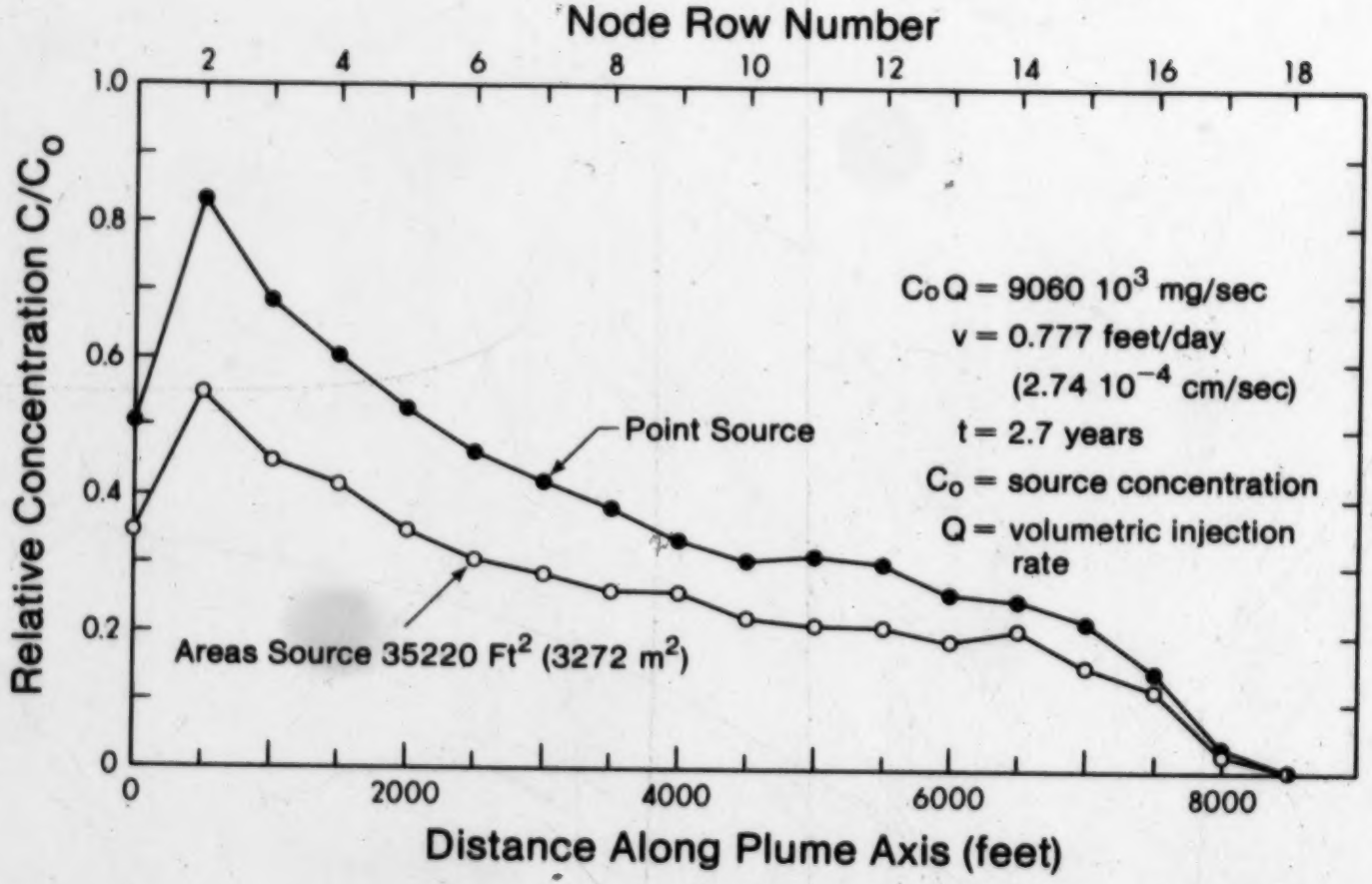
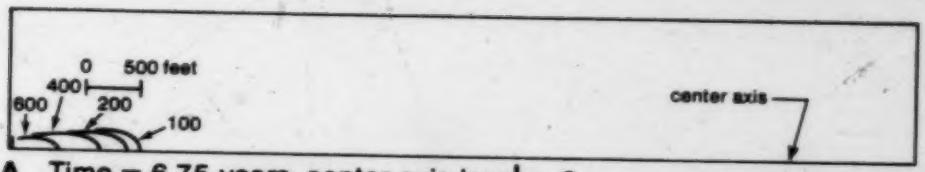
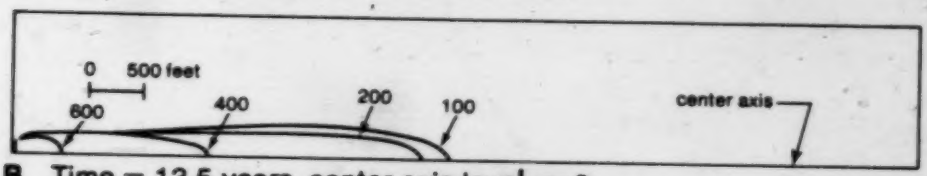


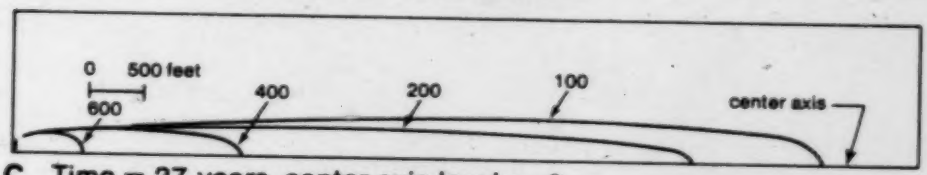
Figure III.9
A given injection rate C_oQ can be obtained by a point source or by an equivalent area source. A considerably more important dilution occurs when an areal source is used as shown by the plot for both point and area sources.



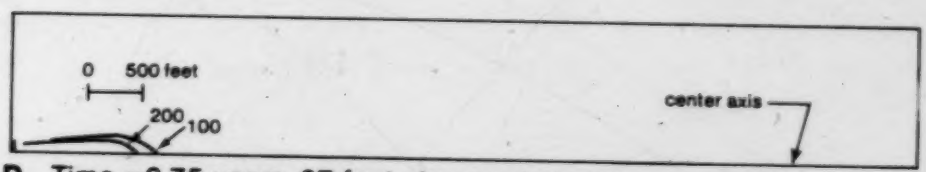
A Time = 6.75 years, center axis level z=0



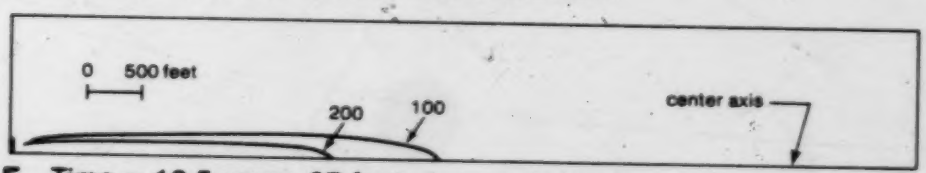
B Time = 13.5 years, center axis level z=0



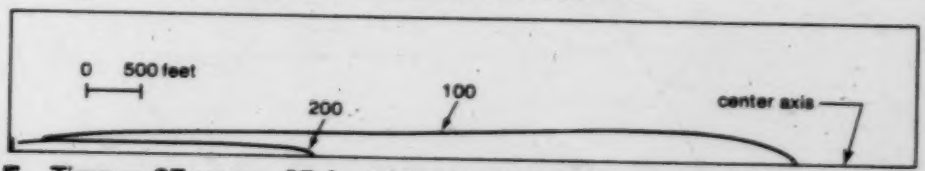
C Time = 27 years, center axis level z=0



D Time =6.75 years, 67 feet above axis level



E Time = 13.5 years, 67 feet above axis level



F Time = 27 years, 67 feet above axis level

Figure III.10
This figure depicts the plume shape in two horizontal planes at two depths and three different times: **A B** and **C** show the plume horizontal spread at the center axis elevation for times of 6.75; 13.5 and 27 years respectively; **D, E** and **F** show the horizontal spread 67 feet above the center axis for times of 6.75, 13.5 and 27 years. Contours are in milligram per liter and C_o=1000 mg/ℓ.

concentration patterns at different depths. Results are shown for the same time periods at an elevation of 67 feet above the center axis (D, E and F).

The accuracy of the numerical solution can be analysed more closely by comparing the analytical and numerical values of concentration along the center axis (Figure III.11) and along transverse axes at various locations (Figure III.12). Results are shown for a simulation using 16 time steps with 2000 particles added per step for a total of 32000.

The direct comparison of numerical results and analytical values for the centerline concentration shows an offset between two nearly parallel curves. This offset is caused by the fact that the analytical solution is in fact the concentration at points along the centerline whereas numerical values indicates concentrations that are averaged over the volume of each element. For this model trial, the center-axis passes through the middle of a central row of elements. Because concentration along this axis is a local maximum, the concentrations, averaged within the element volumes, are less than the analytical ones. A similar condition occurs across the entire plume because the sharp concentration gradient is averaged in a stepwise manner between elements.

Theoretically it would be necessary to integrate the analytical solution over each element and then divide the result over their volume to get results comparable to those

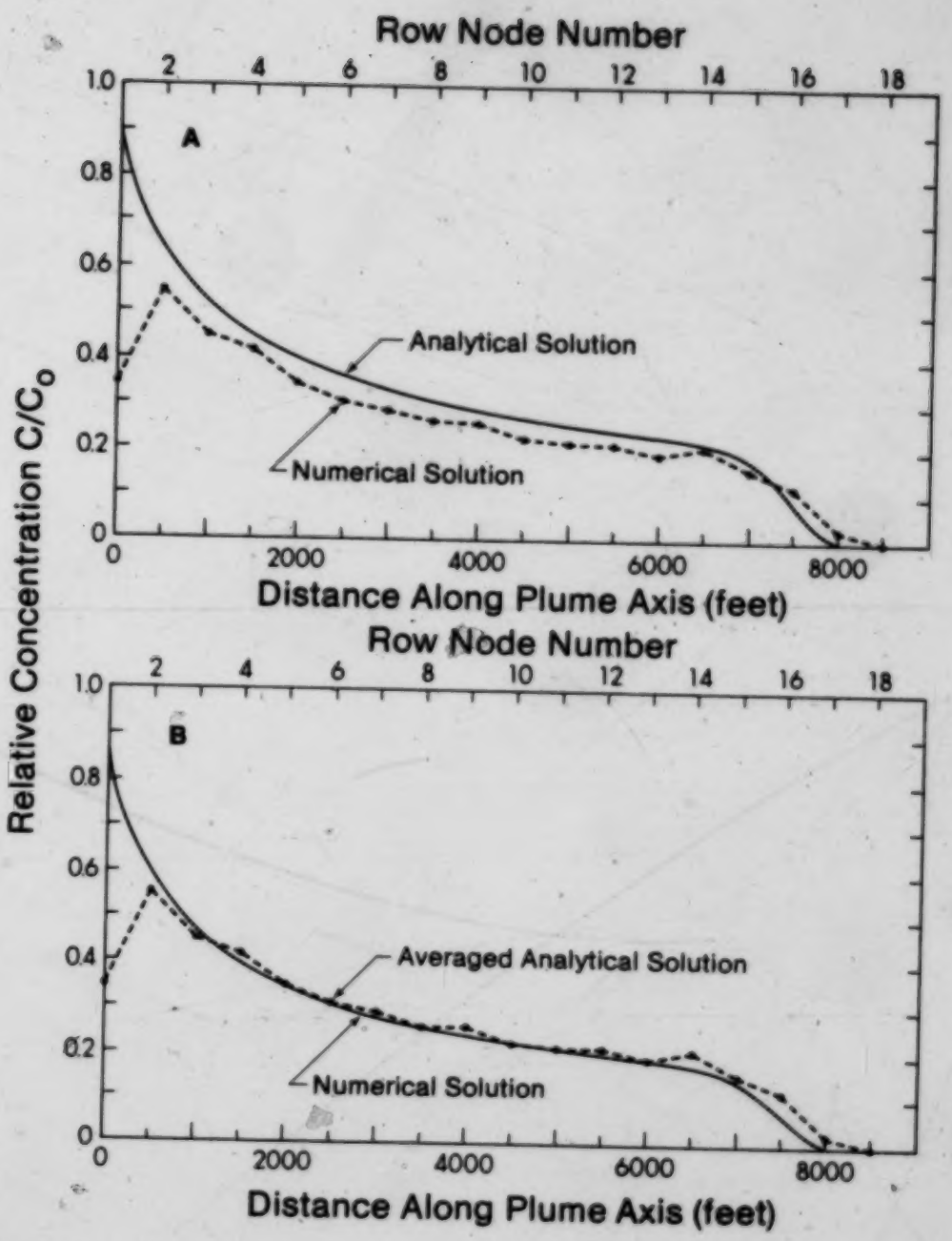


Figure III.11

The numerical solution is obtained by dividing the total particle mass in any element by its volume. The analytical solution gives a value for each point, but can be similarly averaged over element volumes. The analytical concentration at the plume axis is higher than the numerical result (**A**), but the averaged analytical solution gives a more accurate match (**B**). Areal source is same as in Figure III.9.

obtained numerically with 3DFT. In practice, a reasonably accurate approximation is easier to obtain. In fact, it is not necessary to take in account the variation in all three directions. The concentration gradient in the x-direction is approximately constant over the length of each element and the values at mid-element are good representative of averages that would be taken in the x-direction. Moreover because the concentration gradient is relatively more important in the z-direction than in the y-direction, the smaller variations along the y-direction can be neglected. As a result a three-point vertical averaging of the analytical solution is performed at mid-element length with one value taken at center-axis level, the other two at the top and bottom of each element. This analytical average is then compared with the numerical solution in Figure III.11B.

The improved fit indicates the sensitivity of the solution to volumetric averaging. The average deviation from the analytical solution drops from about 6% to less than 2.5% when this averaging is considered. The scatter around the mean or standard deviation remains at 12% in both cases.

Lateral dispersion can be examined using the same technique. Horizontal and vertical concentration patterns are illustrated in Figure III.12 for a distance of 3500 feet (1070m) and 5500 feet (1680m) from the source. The average error is less than 2% in these cases.

The accuracy of the numerical solution varies in direct proportion to the particle density. The total number of

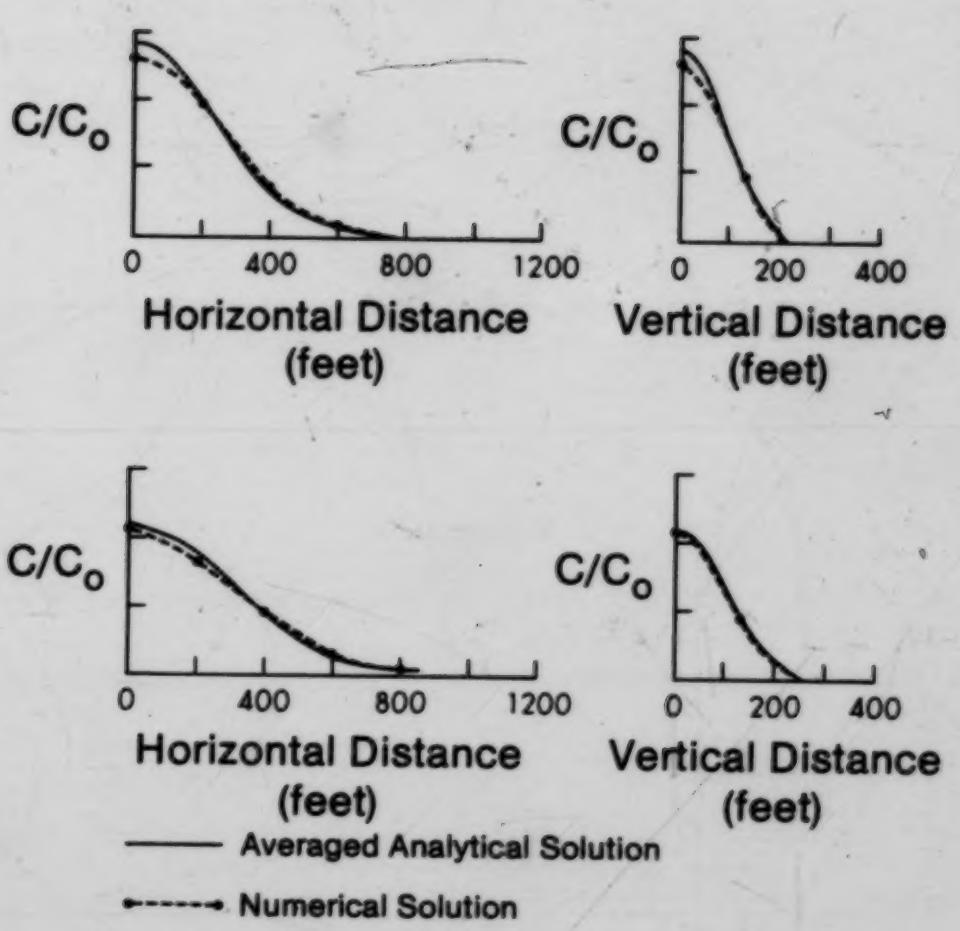


Figure III.12
The lateral spreading of the contaminant is compared with an averaged analytical solution. **A** illustrates the lateral spreading 3500 feet (1070 m)downflow of the source and **B** the lateral spreading at 5500 feet (1680 m).

particles can be increased to reduce the statistical noise. Alternatively an increased density permits the use a finer grid for a more detailed picture of the plume. Figure III.13A shows the results of a trial similar to that of Figure III.11 but using properties of symmetry of the plume around the center axis. Because the plume is symmetrical on each side of the y and z-axes it is possible to simulate only one quarter of the plume with just as many elements and particles. A change in grid scale and source position along the y and z-axis results in a trial equivalent to having a fourfold increase in particle density distributed in four times smaller elements. The results show that the accuracy remains the same with a comparable average error 3% and standard deviation 12%.

The use of a sequence of particle pulses as the approximation of a continuous source is dictated by practical considerations. 3DFT permits to choose any time-step size combined with any number of particles. However, the size of time steps used for input appears to be a sensitive parameter. Smaller time steps produce a much smoother particle input and a better earlier mixing. Comparison of results shows that by cutting time steps by four (Figure III.13B) the average error can be reduced by close to a factor 10, from 3% to 0.4% while the standard deviation is cut by a factor four from 12% to 3% at the expense of a nearly fourfold increase in cost. The choice of a time step should therefore be made from consideration of

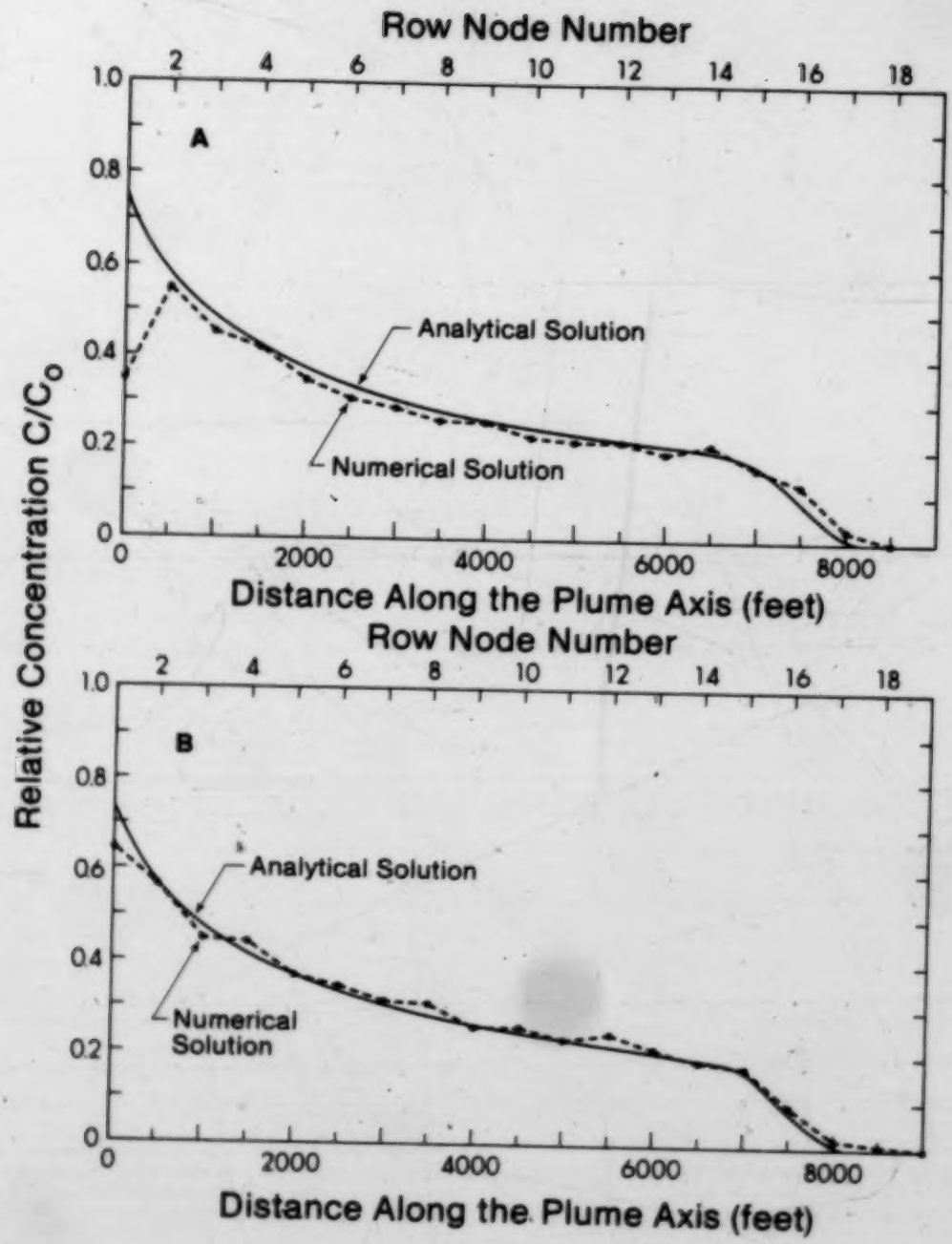


Figure III.13
The sensitivity of the numerical solution to the discrete character of the source function can be illustrated by comparing the results with a source using **A**: a time step of 1.68 years and 2000 particles per step, and **B**: a time step of 0.42 years and 500 particles per step.

the largest time step giving an acceptable accuracy.

D. Practical Constraints in Applying the Code

The development of 3DFT is motivated by the need for a simulation tool aimed specifically at practical applications. It is important to realize however that the model has theoretical and practical constraints which limit its range of applicability. From a theoretical viewpoint, applications are restricted to saturated-flow domains limited by non-deformable boundaries. The principal directions of permeability are assumed to be parallel to element sides so that oblique anisotropy is not handled accurately. Also, density-dependent flow cannot be simulated using the actual code because concentration gradients are assumed not to affect the flow patterns. As a first approximation dispersion is assumed to be Fickian and isotropic with respect to permeability. Large scale dispersivities can in principle be handled through the consideration of geological heterogeneities.

In theory the model is applicable to a vast array of problems. Some identified in chapter one included contaminant problems related to the infiltration of tailing leachates, seawater intrusion, chemical spills and sanitary landfills. The important practical constraints on the application of the approach are based on the numerical methods and on the discrete representation of the flow domain. Lets look at these constraints in more detail.

The maximum size of the grid is limited by workspace constraints. The limiting parameters are the total number of nodes with unknown hydraulic heads and the bandwidth. The maximum storage allocated to the workspace is one megabyte or 125000 double precision entries. With those constraints some of the largest grids could be of 26X26X2 nodes for a thin square domain, 10X10X9 for a cube and 20X10X6 for an intermediate shape. In every cases the storage requirement is close to the maximum and no extra layers, columns or rows can be added.

Larger grids could be used if necessary with the help of special assembler-language subroutines giving access to a larger storage for the workspace. This storage can be increased to a maximum of 6 megabytes at the expense of a proprotional increase in cost. The largest grids in this case would be 41X41X2 for the thin square domain, 13X13X12 for the cube and 32X13X7 for the intermediate-shape.

This constraint on the number of elements results on the maximum grid size and means that thin beds or planar features such as fractures are difficult to include because the elements in the code must be uniform in size. This constraint, caused by the transport solver, in turn forces the averaging of properties over the element volumes. A variable-density grid would be a more appropriate device helping to deal with these factors.

In addition the relatively coarse representation of the plume resulting from the relatively small number of the

elements can produce a staircase image with the smallest unit being the same size as the element.

IV. CONCLUSIONS AND RECOMMENDATIONS

This thesis has documented the development and successful testing of a three-dimensional model for the simulation of flow and mass-transport in groundwater systems. A segmented code together with user-defined options permit the convenient testing of the program and the display of various results. These features are helpful to debug the program and to set up the data correctly. The overall procedure involved in using the program is facilitated by the use of an automatic mesh generator combined with a concise format for data input based on integer maps. Because a minimum number of parameters are needed, it is very simple to generate and modify three-dimensional meshes and to solve for flow and transport for a great variety of boundary conditions.

The approach used for the solution to the flow equation combines a finite-element technique with a direct solution to the resulting system of linear equations. The computational effort is minimized by using a direct sparse-matrix solver. This approach is very cost effective because of the elimination of numerical operations. Storage problems however related to the use of a large workspace remains a major obstacle to the effective and economical handling of large three-dimensional systems. The bandwidth increases very fast with the size of three-dimensional grids. Hence, important limitations remains relative to the extent and complexity of flow domain that can be simulated. Moreover, added costs are associated with the memory

requirements which in turn reduces the feasibility for simulating large systems. Indications are that further cost reductions are possible using an array processor for the execution of large problems. Also, different matrix solvers not requiring any workspace could prove to be more cost effective.

The accuracy of the flow code has been demonstrated for an array of boundary conditions. Although the finite-element approximation to the flow equation is sensitive to high gradients and rapidly changing heads, such sensitivity is typical of numerical solutions in general and does not affect 3DFT's solutions in particular. For example, comparison of 3DFT with Verge's (1975) code shows that the accuracy and numerical behavior are nearly identical.

It is possible to solve for mass transport in three dimensions using a moving-particle method. The use of a direct-simulation approach to mass transport has proven to be simple and versatile. There is no limitations with respect to the element size or time-step duration that can be used but the accuracy of the solution is sensitive to the number of particles and the size of the time steps.

The volume of an individual element can be relatively large but the use of larger elements yields a coarser representation of the contamination plume. Conversely, smaller elements tends to provide a finer description but they produce increased statistical noise because smaller numbers of particles end up in each cell. Indications are

that the number of particles required to maintain a given degree of accuracy is inversely proportional to element volume. Therefore the use of smaller elements must be compensated for by increasing the number of particles in the simulation.

Small transport time-steps are needed for a thorough mixing of individual particle pulses into a continuous plume. The determination of the appropriate number of particles and the size of a convenient time steps is problem-dependent and requires some experimentation by the model user. The cost of trials is, in general, directly proportional to the total number of particles used and to the number of time steps.

Further improvements in the approaches used for 3DFT are desirable and some recommendations can be made with this respect.

(1) The use of an iterative solver for the flow equation, may be a valuable alternative to direct solvers, as it avoids the need for a workspace. The efficient solution to sparse algebraic systems is at present an extremely active area of research in mathematics and computer science. The development of methods for solving sparse linear systems, both directly and iteratively, is a rapidly developing field. Indications are however that iterative solvers would probably be the most appropriate for three-dimensional problems (Carey and Oden, 1984)

(2) A usefull addition to the code could be an optimizing algorithm for node numbering to reduce the bandwidth of the grid. Such an algorithm would allow a more efficient use of direct solvers and thus facilitate the treatment of larger problems than is presently possible with the existing method of node numbering.

(3) An important area for future research related to three-dimensional models is in the development of model pre-processors. These models would be capable of generating variable-density grids and handling available geologic data as well as hydraulic parameters. Such pre-processors could be used to generate complex grids, and assign parameter to the elements and boundary conditions to the nodes on the basis of raw field information.

(4) The implementation of the code with capabilities for handling a variety of chemical processes is an obvious area of future research. The most important processes, radioactive decay and ion-exchange, could be added to the code in a relatively straightforward manner and could greatly inhance the applicability of the actual code.

(5) The actual application of 3DFT to a number of field studies could help to suggest modifications for further improvement the code's general capabilities.

BIBLIOGRAPHY

AHLSTROM S.W., FOOTE H.P., ARNETT R.C., COLE C.R. and SERN R.J.(1977), *Multicomponent Mass Transport Model: Theory and Numerical Implementation (Discrete-Parcel-Random-Walk Version)*, Battelle PNL, Report 2127, 134 pages.

ANDERSON M.P.(1984), Movement of Contaminants in Groundwater: Groundwater Transport-Advection and Dispersion, *Groundwater Contaminants*, Studies in Geophysics, National Academy Press, p.37-45.

APPEL C.A. and BREDEHOEFT J.D.(1976), Status of Ground-Water Modeling in the U.S. Geological Survey, *Geol. Survey Circular 737*, 9 pages.

ASCE (1973), Engineering Computer Program Documentation Standards, *Proceedings of the American Society of Civil Engineering*, v.99, SM3.

BEAR J.(1972), *Dynamics of Fluids in Porous Media*, American Elsevier, N.Y., 774 pages.

BREDEHOEFT J.D. and PINDER G.F.(1973), Mass Transport in Flowing Groundwater, *Water Resources Research*, v.9, no.1, p.194-210.

CAREY G.F. and ODEN J.T.(1984), *Finite Elements Volume III, Computational Aspects*, Prentice-Hall, N.J., 350 pages.

CHAMBERLIN T.C.(1885), Requisite and Qualifying Conditions of Artesian Wells, *U.S. Geological Survey Annual Report 5*, p.131-175.

CHORLEY D.W. SCHWARTZ F.W. and CROWE A.S.(1982) *Inventory and Potential Applications of Groundwater Flow and Chemestry Models*, Research Management Division, Alberta Environment Report RMD 82/7, 73 pages.

DARCY H.(1856), *Les fontaines publiques de la ville de Dijon*, Victor Dalmont, Paris, 647 pages.

DILLON R.T., LANTZ R.B. and PAHWA S.B.(1978), *Risk Methodology for Geologic Disposal of Radioactive Waste : The Sandia Waste Isolation Flow and Transport (SWIFT) Model*, Sandia Corp., Report SAND78-1267, 116 pages.

DOMENICO P.A. and ROBBINS G.A.(1984a), A Dimension Scale Effect in Model Calibrations and Field Tracer Experiments, *J. Hydrology*, no.70, p123-132

DOMENICO P.A. and ROBBINS G.A.(1984b), *The Displacement of Connate Water from Aquifers*, on press.

EISENSTAT S.C., GURSKY M.C., SCHULTZ M.H. and SHERMAN A.H.(1982), I. The Symmetric Codes, *International Journal for Numerical Methods in Engineering*, v.18, p.1145-1151.

FORSCHEIMER P.(1886), Uber die Ergiebigbeit von Brunnenanlagen und Sickerschlitzen, *Zeitschr. Architekten und Ingenieurvereins (Hannover)*, 32, p.539-564.

FORSCHEIMER P.(1898), Grundwasserspiegel bei Brunnenanlagen, *Zeitschr. Architekten und Ingenieurvereins (Hannover)*, 50, p. 629-635, 645-648.

FREEZE R.A. and BACK W.(1983), *Physical Hydrogeology*, Benchmark Papers in Geology, vol.72, Hutchinson Ross Pub. Co. Stroudsberg, Penn., 415 pages.

GARDNER A.O., PEACEMAN A.L. and POZZI R.(1964), Numerical Calculation of Multi-Dimensional Miscible Displacement by the Method of Characteristics, *Society of Petroleum Engineers Journal*, 4(1), p.26-38.

GRAY W.G. and HOFFMAN J.L.(1983), A Numerical Model Study of Groundwater Contamination from Price's Landfill, New-Jersey, *Groundwater*, v.21, no.7, p.7-21.

GUYMON G.L., SCOTT V.H. and HERRMAN L.R.(1970), A General Numerical Solution of the Two-Dimensional Diffusion-Convection Equation by the Finite Element Method, *Water Resources Research*, v.6, no.6, p.1611-1616.

LAWSON D.W.(1971), Improvements in the Finite-Difference Solution of Two-Dimensional Dispersion Problems, *Water Resources Research*, v.7, no.3, p.721-725.

MERCER J.W. and FAUST C.R.(1981), *Groundwater Modeling*, National Water Well Association, Reston, Virginie, 60 pages.

NEUMAN S.P.(1979), A Summary of Computer Methods for Solving the Dispersion-Convection Equation, in *Progress and Problems in the Analysis and Prediction of Sub-Surface Mass-Transport*, (Ed. E.S. Simpson), unpublished Topical Report to the Nuclear Regulatory Commission by the Department of Hydrology and Water Resources, Univ. of Arizona, Tucson, p.78-83.

OGATA A. and BANKS R.B.(1961), A Solution to the Differential Equation of Longitudinal Dispersion in Porous Media, *U.S. Geological Survey Prof. Paper 411-A*, 7 pages.

PINDER G.F.(1973), A Galerkin Finite Element Simulation of Groundwater Contamination on Long Island, New-York, *Water Resources Research*, v.9, no.6, p.1657-1669.

PINDER G.F. and BREDEHOEFT J.D.(1968), Application of the Digital Computer Techniques for Groundwater Resource Evaluation, *Water Resource Res.*, vol.4, p.1069-1093.

PINDER G.F. and COOPER H.H.(1970), A Numerical Technique for Calculating the Transient Position of the Saltwater Front, *Water Resources Research*, v.6, no.3, p.875-882.

PINDER G.F. and GRAY W.G.(1977), *Finite Element Simulation in Surface and Subsurface Hydrology*, Academic Press, N.Y., 295 pages.

PRICKETT T.A. and LONNQUIST C.G.(1971), *Selected Digital Computer Techniques for Groundwater Resource Evaluation*, Illinois State Water Surv. Bulletin 55, 62 pages.

PRICKETT T.A., NAYMIK T.G. and LONNQUIST C.G. (1981), *A "Random Walk" Solute Transport Model for Selected Groundwater Quality Evaluations*, Illinois State Water Survey, Champaign, Bulletin 65, 103 pages.

REMSON I. HORNBERGER G.M. and MOLTZ F.J.(1971), *Numerical Methods in Subsurface Hydrology*, Wiley-Interscience, N.Y., 389 pages.

ROBBINS G.A. and DOMENICO P.A.(1984), Determining Dispersion Coefficients and Sources of Model-Dependent Scaling, *Journal of Hydrology*, no.75.

RUSHTON K.R. and REDSHAW S.C.(1979), *Seepage and Groundwater Flow*, Numerical Analysis by Analog and Digital Methods, John Wiley &Sons, N.Y., 339 pages.

SEGOL G.A.(1977), A Three-Dimensional Galerkin Finite Element Model for Analysis of Contaminant Transport in Saturated-Unsaturated Porous media, in *Finite-Elements in Water Resources*, Gray, W.G., Pinder, G.F. and Brebbia, C.A., Pentech Press, London.

SCHWARTZ F.W.(1977), Macroscopic Dispersion in Porous Media: The Controlling Factors, *Water Resources Research*, v.13, no.2, p.743-752.

SCHWARTZ F.W.(1978), Applications of Probabilistic-Deterministic Modeling to Problems of Mass Transfer in Groundwater Systems, in *Third International Hydrology Symposium*, Ft. Collins, p.281-296.

SCHWARTZ F.W.(1984), *Modeling Groundwater Flow and Composition*, Applied Geochemestry Workshop, unpublished.

SCHWARTZ F.W. and CROWE A.(1980), *A Deterministic-Probabilistic Model for Contaminant Transport*, U.S. Nuclear Regulatory Commission, NUREG/CR-1609, 158 pages.

SCHWARTZ F.W. and DONATH F.A.(1980), *Scenario Development and Evaluation Related to the Risk Assessment of High-Level Radioactive Wastes Repositories*, Rep.NUREG/CR-1609, 158 pp., U.S. Nucl. Regul. Washington, D.C..

SCHWARTZ F.W. and SMITH L.(1981), An Evaluation of the Ability to Predict Radionuclide in Groundwater Flow Systems, in *Scientific Basis for Nuclear Waste Management*, v.3, (Ed. J.Moore), Plenum Publishing Corp., p.599-605.

SHAMIR U.Y. and HARLEMAN R.F.(1967), Numerical Solutions for Dispersion in Porous Media, *Water Resources Research*, v.3, no.2, p.557-581.

SLICHTER C.S.(1898), Theoretical Investigation of the notion of Groundwaters, *U.S. Geological Survey Annual Report 19*, p.295-384.

SMITH L. and SCHWARTZ F.W.(1980), Mass Transport 1. A Stochastic Analysis of Macroscopic Dispersion, *Water Resources Research*, v.16, no.2, p.303-313.

SMITH L. and SCHWARTZ F.W.(1981), Mass Transport 2. Analysis of Uncertainty in Prediction, *Water Resources Research*, v.17, no.2, p.351-369.

STRAUTMAN S.Y.(1984), *Determining Dispersion Coefficients from a Landfill Leachate Plume*, M.Sc. Dissertation, Texas A&M University.

THEIS C.V.(1935), The relation Between the Lowering of the Piezometric Surface and the Rate and Duration of Discharge of a Well Using Ground-Water Storage, *Am. Geophys. Union Trans.*, no. 16, p519-525.

THIEM G.(1906), *Hydrologische Methoden*, Gephardt, Leipzig, 56 pages.

VERGE M.J.(1975), *A Three-Dimensional Saturated-Unsaturated Groundwater Flow Model for Practical Applications*, Ph.

D. Dissertation, Waterloo University, 242 pages.

WANG H.F. and ANDERSON M.P.(1982), *Introduction to Groundwater Modeling Finite Difference and Finite Element Methods*, Freeman and C., S.P., 237 pages.

WENZEL L.K.(1942), Methods of Determining Permeability of Water-Bearing Materials with Special Reference to Discharging-Wells Methods, *U.S. Geol. Survey Water-Supply Paper 887*, 192 pages.

APPENDIX 1. 3DFT User's Manual

Table of Contents

1. INTRODUCTION

2. DESCRIPTION OF THE COMPUTER MODEL

2.1 Model Structure

2.1.1 Main Program
2.1.2 Subroutine QUAD
2.1.3 Subroutine MESH
2.1.4 Subroutine MASTR
2.1.5 Function VELO
2.1.6 Subroutine DISPER
2.1.7 Subroutine SDRV
2.1.8 Subroutine SOLVE
2.1.9 Subroutine AREA
2.1.10 Subroutine INPUT
2.1.11 Subroutine INPUMP
2.1.12 Subroutine INTRA
2.1.13 Subroutine PATCH
2.1.14 Subroutine OUT1
2.1.15 Subroutine OUT2
2.1.16 Subroutine OUTPUT
2.1.17 Subroutine MAP
2.1.18 Subroutine RANDU
2.1.19 Function ZERO

2.2 Input and Output Operations

2.2.1 Input
2.2.2 Output

3. INPUT PARAMETERS AND VARIABLES

3.1 Input Variables
3.2 Input Card Instructions
3.3 Dimensioning of the Arrays

4. MODE OF OPERATION

4.1 Array Dimensioning
4.2 Compilation
4.3 Simulation Run

1. INTRODUCTION

The purpose of this manual is to describe the Three-Dimensional Flow and Transport (3DFT) model. A description of the program is presented in Section 2. Section 3 is a detailed examination of the input and output parameters. Section 4 explains the sequence of operations needed to complete a simulation.

2. DESCRIPTION OF THE COMPUTER CODE

This section presents a brief description of the model. The first part includes a discussion of the program's structure. The second part presents a description of input and output operations.

2.1 Model Structure

The basic sequence of operations of the model is illustrated in Figure 1. First, all input data are read and printed to provide the user with a data echo. Data required for a flow simulation include a description of the region, boundary conditions and hydraulic parameters. Data for the transport problem include source area specification, the chemical and physical properties of the contaminant and parameters to control the simulation. In the next step the nodes, coordinates and elements are generated for the three-dimensional finite element grid. Values of permeability, coefficient of storage, porosity and dispersivity are assigned to the appropriate elements.

Following this step, a three-dimensional finite element model, based on the Galerkin formulation, is used to simulate the steady state head distribution in the region of flow for a steady or transient flow regime, equation (1).

The finite-element method provides a direct solution to equation (1) by approximating the differential form by smaller, simpler equations known as shape functions. The shape functions used here are linear because the shape of the element side is linear (i.e. 2 nodes per edge).

Finally the mass transport problem is solved by tracking reference particles with their associated mass through the porous medium. A specified number of particles are added to the region at the contaminant source at the beginning of each time-step. By varying the number added during a time-step, it is possible to approximate even very variable contaminant leaching input functions.

At the beginning of each time-step, a velocity for each reference particle is obtained by linear interpolation from the surrounding nodal head values, substituted in Darcy equation, together with hydraulic conductivity values and porosity. Each particle moves to an interim position along

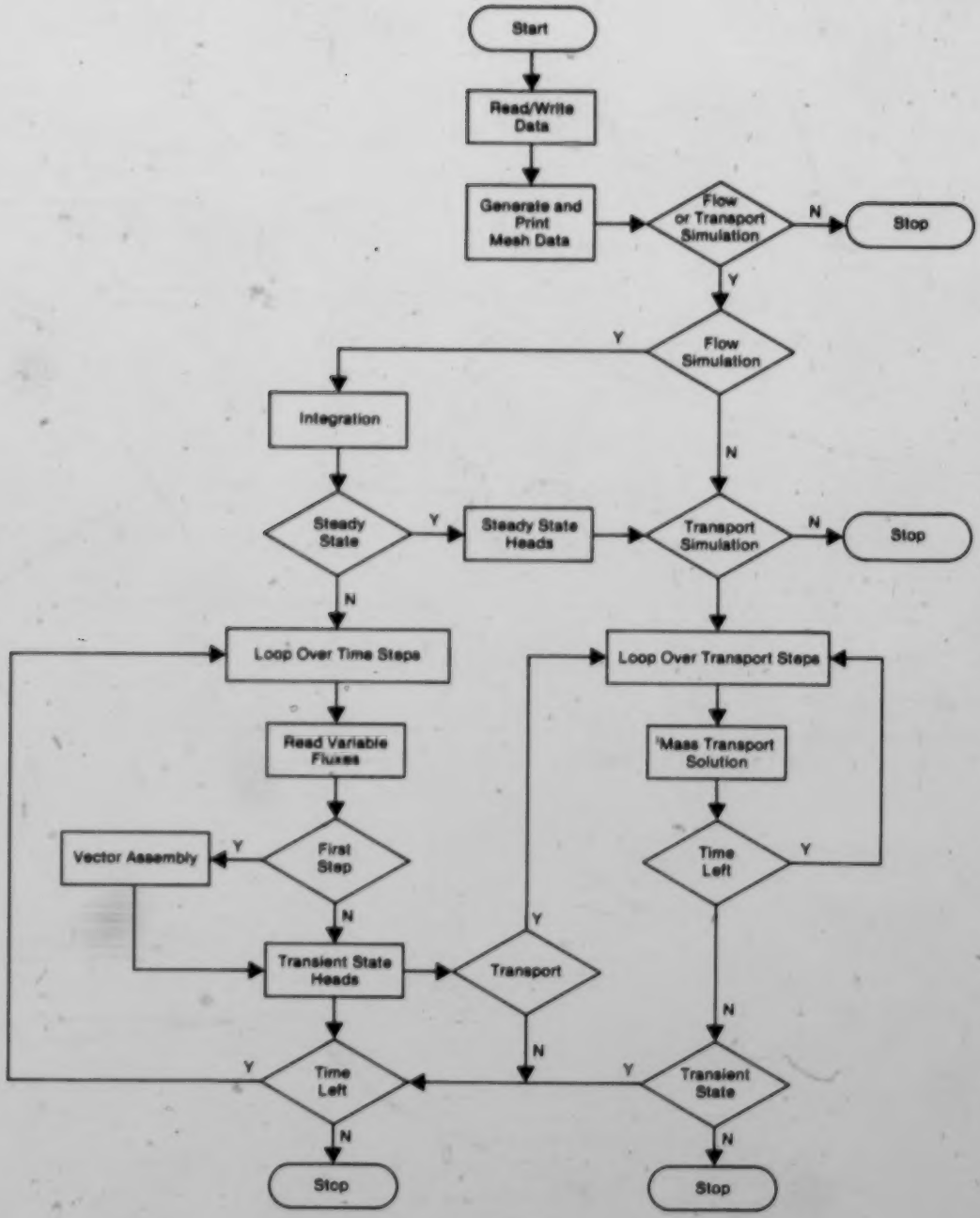


Figure 1. Overview of 3FDT's main program

respective vectors for a distance that is determined by the magnitude of the time-step. The particle is relocated by the random component of motion in the vicinity of the interim position. The model is designed to loop through the number of time-steps defined by the user. At each time-step, a new particle distribution is calculated.

The model was programmed in FORTRAN IV and run and tested using the AMDAHL 5860 computer at the University of Alberta. Basically, the computer program consists of twenty-two parts; the main program, sixteen subroutines, two functions and a package of four subroutines used as a matrix solver. These segments are discussed in detail below.

2.1.1 Main Program

The purpose of the main program essentially is to control the sequence of program steps in the simulation by calling the subroutines. Virtually none of the actual simulation steps are carried out in the main program. The storage required by the arrays is outlined in Section 3. For simplicity, it is only necessary to specify array sizes in a separate common block file. A more specific description of the input necessary for a simulation is discussed in the following sections.

2.1.2 Subroutine QUAD

Subroutine QUAD computes the coefficients of the linear system of equations needed to solve for the steady (or transient) state hydraulic head distribution over the finite element grid. The subroutine requires the input of the finite element grid and the hydraulic conductivity (and storativity) of the elements both from subroutine MESH, the location of the water table and any constant head or flux values (also from subroutine MESH). Values of hydraulic head are calculated by sequencially progressing through the finite elements in the grid. Because we use a compressed mode of storage, the configuration of the grid surrounding a node, or patch, is of importance for the allocation of storage in the arrays containing the coefficients. Thus, it has to be known before each integration step. With this information, coming from subroutine PATCH, both conductivity and storativity matrices can be assembled.

The 8 by 8 elemental matrices are assembled individually and loaded in global vectors by keeping track of the rows and columns location of the coefficients. The flux vector, which describes the flow across element boundaries is then assembled. With these steps completed at the beginning of the flow simulation, the system of linear equation need to be further compressed to get rid of all zero coefficients. Time varying conditions have to be accounted for transient state solutions. These steps are carried out in the MAIN program which also calls the equation solver. The equations are solved by solution driver SDRV to obtain the values for the unknown hydraulic head. SDRV is called by subroutine SOLVE which expands the head vector by inserting the known constant head values in the proper sequencial order, and assembles the linear system in its final form to be solved.

2.1.3 Subroutine MESH

A tri-linear finite element model is used to calculate the head distribution at the nodal points over the grid. The finite element grid is regular (i.e. constant node spacing in the coordinate directions). Because such a simple grid is easily generated within the model, the data necessary for a simulation is reduced considerably. The only information required to construct the finite element grid is the number of rows, columns and layers or levels of nodes, the spacing between nodes in the three main directions and the layer number of each top node defining the position of the water table.

The subroutine first fills three vectors with the coordinates of the nodes and a three-dimensional array (the incidence matrix) is loaded with the spatial arrangement of numbered nodes. Nodes are numbered consecutively from 1 starting at the origin of the coordinates moving upwards to the position of the water table, and then back to the bottom repeating the procedure along the width of the grid repeating again for every node along the grid length. Constant head and flux nodes are designated as such within the program, in separate vectors.

Following this, the elements are constructed and numbered sequencially in a way similar to the nodes. Element incidences are determined from the node numbers stored in the three-dimensional incidence matrix. For this purpose the standard procedure is applied to the right handed cartesian coordinate system: counter-clockwise from bottom to top. Finally, each element is attributed geologic parameters.

2.1.4 Subroutine MASTR

Subroutine MASTR controls the mass transport portion of the model. The subroutine is designed to move the contaminant particles through the groundwater system according to the probabilistic-deterministic modelling techniques.

Initially, particles are randomly placed within an area, the position of the particles being defined by X, Y and Z coordinates. Particles in the entire system are moved one at a time during any one transport time-step. First, components of the seepage velocity at the particle location are determined from function VELO and the particle is moved with a deterministic motion. If the dispersive character of the transport system requires, the particle is randomly relocated by calling subroutine DISPER.

Concentration is determined by summing the total mass of contaminant in the cell and dividing by the volume of water in the cell. This sequence of operations is repeated for each particle in the system, after which new particles are added to the source cell at the start of the next simulation cycle.

A particle transport feature is available that designates sink nodes around which incoming particles are effectively removed from further transport within the model. The distribution of particles and the concentration are determined and may be output to the line printer using subroutine MAP.

2.1.5 Function VELO

The movement of the contaminants is controlled primarily by the velocity of groundwater flow. The velocity at a point is generated in this subroutine from the head values and from conductivity and porosity.

Velocity values in all three coordinate directions are calculated for each particle location by interpolation between 8 surrounding internode velocities. For interpolation purposes, it is assumed that the velocity components, outside and perpendicular to a no-flow boundary (dummy velocities), are equal in magnitude to those inside and near the boundary but opposed in direction. For constant

head boundaries dummy velocities are assumed to be of the same magnitude and direction as the internal perpendicular components. Thus, particles are reflected back across the no-flow boundary and into the groundwater flow system. Also, along a discharge or recharge boundary, the velocity values are assumed to be equal to those in the adjacent elements, thereby allowing particles to leave the region.

2.1.6 Subroutine DISPER

This subroutine determines the random motion of the particles. The magnitude of the motion is determined by the dispersivity, the seepage velocity, diffusion coefficients, time step size and by uniformly distributed random numbers. The position of the particle is checked to determine whether or not it has encountered any boundary conditions (such as a no-flow boundary or discharge to the surface). Depending upon the conditions involved, the particle may be relocated.

2.1.7 Subroutine SDRV

The solution driver SDRV uses a form of the LU decomposition, known as the Cholesky method, to decomposes the global conductivity matrix, and a backsubstitution to solve for the unknowns in the system of linear equations and thus to yield the solution for the hydraulic head distribution. This decomposition method is applied to matrices that are symmetrical and sparse. SDRV breaks the solution in three steps each corresponding to a different subroutine (SSF, SNF and SNS). The special feature of this particular subroutine package is that it is designed to work with sparse matrices.

2.1.8 Subroutine SOLVE

This subroutine prepares the arrays of the linear system of equations for the sparse matrix solver (SDRV) according to the nature of the flow simulation.

2.1.9 Subroutine AREA

Once the head distribution is known reference particles are introduced in the flow field. This is done by calling AREA and by specifying the dimensions of the source area at one of the domain's external boundaries. The number of particles to be generated at each transport time step is also required. Particles are initially uniformly placed over the area before being moved in the system.

2.1.10 Subroutine INPUT

Reads the title and input data needed to generate the mesh and to solve for the flow (or transport) problem (input card 1 to 40).

2.1.11 Subroutine INPUMP

Reads from data file the information needed to run the transient state flow simulation (card 41-42).

2.1.12 Subroutine INTRA

Reads the specific data needed to generate particles, to run the mass transport simulation and to print the result (card 43-45).

2.1.13 Subroutine PATCH

Checks the configuration of the grid around a node and eliminates from the patch the contribution of constant head nodes and nodes lying outside the flow domain, such that only the relevant non-zero coefficients of the linear system are stored.

2.1.14 Subroutine OUT1

Sends a data echo to the user via the output device (IOUT) according to the output option selection.

2.1.15 Subroutine OUT2

Prints the mesh generation data. It displays a summary of the finite element mesh features. The complete node and element specifications can also be printed on option.

2.1.16 Subroutine OUTPUT

After the flow equation is solved, head values are printed in a layer-by-layer matrix format to facilitate legibility. Transient results are displayed after each time step with a summary of temporary flux values. Steady state results are saved in a scratch file (device 7) and can be used as initial values for a transient state simulation.

2.1.17 Subroutine MAP

This subroutine outputs the particle distribution and concentration and indicates the elapsed simulation time.

2.1.18 Subroutine RANDU

The movement of the reference particles is controlled by deterministic and probabilistic components of particle motion. One requirement for the calculation of the probabilistic motion is a uniformly distributed random number. This subroutine generates pseudo-random numbers in the range of 0 to 1 by the overflow of a single precision integer variable. It is machine dependent, suitable for use with computers that have a 32 bit word, for example the IBM and AMDAHL. Otherwise, it may be necessary to modify it.

2.1.19 Function ZERO

Checks against zero divide and makes sure some dividers have an absolute value of at least 1E-30.

2.2 Input and Output Operations

2.2.1 Input

Data required as input is discussed in this section. A list of input variables and paramaters and the input card setup is presented in Section 3.

Nineteen program control parameters (IOP) are used in the program to control the nature of the simulation and the type of output required. If the user wishes to choose one of the particular functions, IOP should be set to 1 (a value of 0 indicates that the routine or output is not required). The user has the option of running various routines, such as:

IOP(16): Steady(0) or transient state(1) simulation
IOP(17): Read out initial heads(1), otherwise initialized at zero.
IOP(18): Use backward(0) or centered(1) time difference
IOP(19): Run mass transport, print results

The finite element mesh is generated automatically. Several groups of calculations may be output to a line printer:

IOP(1) : Mesh specification summary
IOP(2) : Specified heads and fluxes in the grid
IOP(3) : Geologic code
IOP(4) : Node types and coordinates
IOP(5) : Element and grid incidences
IOP(6) : Element conductivities, specific storage & porosity

Also, the following information can be printed as an aid to program debugging:

IOP(7) : Element conductivity matrix and determinant
IOP(8) : Ordered element incidences
IOP(9) : Node patches
IOP(10): JA,A and flux vector entries as loaded
IOP(11): A,JA,IA vectors
IOP(13): Flux vector

The grid for the model is a regular array of nodes and elements. This grid is constructed within the program according to the user's input parameters. The user must specify the number of rows (NODX), columns (NODY) and layers (NODZ) of nodes and the node spacing in all the three directions (DELX, DELY and DELZ). Any consistent set of

units can be used for mass spatial dimension and time (M, L, T).

The grid may contain up to 30 different hydraulic conductivity (NCOX, NCOY, NCOZ), storativity (NSTO), porosity (NPOR) and transport units (NTU). The units are set up by assigning a numbered code, from 1 to 29 (ICOX, ICOY, ICOZ, ISTO, IPOR, and ITU respectively), to each element within that unit. The codes are read in one line at a time having one code value assigned to each element on that line. Each line starts a new card. The value of the parameter assigned to a unit (COX, COY, COZ, STO, POR, DISX, DISY, DISZ and DIFU) is input with an accompanying unit number code identifier (I), which corresponds to the unit in the three-dimensional mesh. This procedure is a very simple method for entering the data. As illustrated in the example in Section 4, a simple graphical representation of the grid on a layer by layer basis is formed for the particular variable. This style of data input facilitates rapid entry of data for complex settings, rapid alterations and easy checking of input data.

Several hydrogeological parameters are required to estimate the head distribution.

First, the shape of the region along the upper boundary must be defined by identifying the layer elevation, for each row and column of nodes, that best approximates the position of the water table (IDWT).

Second, some of the external boundary nodes may be assigned constant head values.

There are six sides delimiting the grid extent in three dimensions, one pair of sides for each coordinate axis. Sides referred to as X1 and X2 (side 1 and 2) corresponds to the extension of the grid along the X-axis with side X1 at x=0 and X2 at x>0. A similar relationship holds for the Y-axis defining sides Y1 at y=0 (side 3) and Y2 at y>0 (side 4). The Bottom and Top boundaries corresponds to the grid extension along the vertical Z-axis with BOTTOM at z=0 (side 5) and TOP at z>0 (side 6).

A code sequence is used to indicate which nodes along these boundaries are assigned a constant head value or a specified constant flux value, and which have head values calculated within the program (ITOBC, IX1BC, IX2BC, IY1BC, IY2BC and IBOBC for the TOP, X1, X2, Y1, Y2 and BOTTOM boundaries respectively). A code value of 1 indicates that the node has a constant head value, a value of 2 indicates a

specified flux value; these values are read on the following cards, and 0 indicates that the head at that node will be calculated within the program.

The code sequence is required for all nodes along these six boundaries, whether or not a constant head or flux value is present. The constant head values (VTOCH, VX1CH, VX2CH, VY1CH, VY2CH and VBOCH), and the specified flux values (VTOFL, VX1FL, VX2FL, VY1FL, VY2FL and VBOFL) are read sequencially (if required). The sequence is TOP, X1, X2, Y1, Y2 and BOTTOM sides. Each side or boundary is represented as a rectangular coded map with the geometrical axes origin at the lower left corner as if viewed from the positive side of the remaining axis (in a right handed cartesian coordinate system). The scan over the map is done from the left most column to the right and from the bottom line to the top. One advantage of being able to assign constant head values along all the boundaries is to allow the user the flexibility of simulating a smaller piece of a much larger cross-section in more detail.

Reference particles are introduced to the system within a specified area defined by the boundary number (ISIDE) and the two pairs of numbers giving the coordinate range of the area (ARE(1) to ARE(4)). The number of particles that are added at each time-step (NPART) must be input by the user. It is possible to limit particle motion within parts of the region. A code vector (ISINK) is used to indicate nodes in the vicinity of which particles can be removed from flow (ISINK>0). Once a particle gets within one half-element around a sink node, it is, assumed to be lost from the system. The corresponding vacant entry in the coordinate vector is then used for another particle. The total number of particles removed by a sink at each transport step is printed as well as the cumulated number of particles and the total number of remaining particles.

2.2.2 Output

The simulation model will produce an echo of the basic input data and a variety of optional output. The basic output consists of the problem title; the program control parameters; geological, hydrogeological and mass transport parameters and variables; the maps of the codes for the conductivity, storativity, porosity, and particle transport, as well as the actual values of the parameters assigned to each of the units; and a summary of the element grid parameters.

The optional output includes those values discussed in the previous section, selected by setting IOP to 1. The output for the mass transport simulation represents a summary of the distribution and concentration of the particles at selected time-steps. User selected parameters are available to control the mass transport simulation. If a steady-state flow simulation is run it is necessary to specify the number of transport steps (NTR), the size of these steps (DELR) and the number of steps that pass before the results are printed (NSKIP). When a transient flow regime is simulated it is only required to specify NSKIP and the number of transport steps per flow step (NTR), their duration being calculated.

3. INPUT PARAMETERS AND VARIABLES

3.1 Input Variables

This section provides the user with a list of the input variables and their definitions.

problem identifier:

TITLE : any title up to 80 characters

program control parameters:

IOP(1) : mesh specification summary
IOP(2) : specified heads and fluxes in the grid
IOP(3) : geologic code maps
IOP(4) : node types and coordinates
IOP(5) : element and grid incidences
IOP(6) : element hydraulic properties
IOP(7) : element conductivity matrix and determinant
IOP(8) : ordered element incidences
IOP(9) : node patches
IOP(10): JA,A and flux vector entries as loaded
IOP(11): A,JA,IA vectors
IOP(12): sink information
IOP(13): flux vector
IOP(15): compute and write heads
IOP(16): steady(0) or transient state(1) simulation
IOP(17): read out initial heads
IOP(18): use backward(0) or centered(1) time difference
IOP(19): run mass transport, print results

input parameters:

```
NODX   : number of nodes in the X-direction
NODY   : number of nodes in the Y-direction
NODZ   : number of nodes in the Z-direction
NCOX   : number of units of X-direction conductivity units
NCOY   : number of units of Y-direction conductivity units
NCOZ   : number of units of Z-direction conductivity units
NSTO   : number of storage coefficient units
NPOR   : number of porosity units
NTU    : number of transport units
DELX   : node spacing in the X-direction
DELY   : node spacing in the Y-direction
DELZ   : node spacing in the Z-direction
NB     : estimated bandwidth of the coefficient matrices
NNODE  : estimated number of nodes
NCHEAD : total number of constant head nodes
NICH   : total number of internal constant head nodes
NIFL   : total number of internal specified flux nodes
NSP    : workspace storage dimension
NSINK  : number of sink nodes
PINMAS : mass of each reference particle
RES    : maximum particle displacement (element fraction)
```

mesh parameters:

```
IDWT   : level number (of each row and column) defining the
         water table position.
ITOBC  : code indicating constant head(1) or flux(2)
         water table (top boundary) nodes (0 otherwise)
VTOCH  : constant head values assigned to the water table
         nodes, (L)
VTOFL  : specified flux values assigned to the water table
         nodes, (L/T)
IX1BC  : code indicating constant head(1) or flux(2)
         X1 -boundary nodes (0 otherwise)
VX1CH  : constant head values assigned to the X1 boundary
         nodes, (L)
VX1FL  : specified flux values assigned to the X1 boundary
         nodes, (L/T)
IX2BC  : code indicating constant head(1) or flux(2)
         X2 -boundary nodes (0 otherwise)
VX2CH  : constant head values assigned to the X2 boundary
         nodes, (L)
VX2FL  : specified flux values assigned to the X2 boundary
         nodes, (L/T)
IY1BC  : code indicating constant head(1) or flux(2)
         Y1 -boundary nodes (0 otherwise)
VY1CH  : constant head values assigned to the Y1 boundary
         nodes, (L)
```

VY1FL : specified flux values assigned to the Y1 boundary nodes, (L/T)
IY2BC : code indicating constant head(1) or flux(2) Y2 -boundary nodes (0 otherwise)
VY2CH : constant head values assigned to the Y2 boundary nodes, (L)
VY2FL : specified flux values assigned to the Y2 boundary nodes, (L/T)
IBOBC : code indicating constant head(1) or flux(2) bottom boundary nodes (0 otherwise)
VBOCH : constant head values assigned to BOTTOM boundary nodes, (L)
VBOFL : specified flux values assigned to BOTTOM boundary nodes, (L/T)
N1 : sink number
N2 : node number of sink node
I : unit number
COX : X-direction conductivity assigned to a unit, (L/T)
ICOX : X-direction conductivity unit identifier
COY : Y-direction conductivity assigned to a unit, (L/T)
ICOY : Y-direction conductivity unit identifier
COZ : Z-direction conductivity assigned to a unit, (L/T)
ICOZ : Z-direction conductivity unit identifier
STO : storage coefficient assigned to a unit, (L/L)
ISTO : storage coefficient unit identifier
POR : porosity assigned to a unit, (L3/L3)
IPOR : porosity unit identifier
DISX : longitudinal dispersivity assigned to a unit, (L)
DISY : transverse dispersivity assigned to a unit, (L)
DISZ : transverse dispersivity assigned to a unit, (L)
DIFU : coefficient of diffusion assigned to a unit, (L2/T)
ITU : transport unit identifier
T1 : transient state simulation period start, (T)
T2 : transient state simulation period end, (T)
NPP : number of pumping periods per transient simulation
NW : number of flux nodes for a pumping period
IDF : transient flux node identifier
TFLUX : transient flux value, (L/T)
ISIDE : domain boundary (side) number
ARE : coordinate range for the source area, (L)
NTR : number of transport steps per pumping period
NSKIP : transport steps skipped before printing results
DELR : transport step duration, (T)
NPART : particles generated at each transport step
IDICH : internal constant head node number
VICH : constant head value assigned to internal nodes, (L)
IDIFL : internal specified flux node number
VIFL : specified flux value assigned to internal nodes, (L/T)

3.2 Input Card Instructions

The variables and parameters input into the program must conform to the order, format, and columns of the following cards:

CARD	VARIABLE	FORMAT	COLUMNS
1	TITLE	20A4	1 - 80
2	IOP(1...6)	6I2	1 - 12
3	IOP(7...11)	5I2	1 - 10
4	IOP(12...18)	7I2	1 - 12
5	IOP(19)	I2	1 - 2
6	NODX	I5	1 - 5
	NODY	I5	6 - 10
	NODZ	I5	11 - 15
	NCOX	I5	16 - 20
	NCOY	I5	21 - 25
	NCOZ	I5	26 - 30
	NSTO	I5	31 - 35
	NPOR	I5	36 - 40
	NTU	I5	41 - 45
7	DELX	E10.3	1 - 10
	DEYZ	E10.3	11 - 20
	DELZ	E10.3	21 - 30
	NB	I5	31 - 35
	NNODE	I5	36 - 40
	NCHEAD	I5	41 - 45
	NICH	I5	46 - 50
	NIFL	I5	51 - 55
8	NSP	I10	1 - 10
	NSINK	I10	11 - 20
	PINMAS	E10.3	21 - 30
	RES	E10.3	31 - 40
9	IDWT	35I3	1 - 3,4 - 6,...
10	ITOBC	80I1	1...80
11	VTOCH	8F10.0	1 - 10,11 - 20,...
12	VTOFL	8E10.3	1 - 10,11 - 20,...
13	IX1BC	80I1	1...80
14	VX1CH	8F10.0	1 - 10,11 - 20,...
15	VX1FL	8E10.3	1 - 10,11 - 20,...
16	IX2BC	80I1	1...80
17	VX2CH	8F10.0	1 - 10,11 - 20,...
18	VX2FL	8E10.3	1 - 10,11 - 20,...

19	IY1BC	80I1	1...80
20	VY1CH	8F10.0	1 - 10,11 - 20,...
21	VY1FL	8E10.3	1 - 10,11 - 20,...
22	IY2BC	80I1	1...80
23	VY2CH	8F10.0	1 - 10,11 - 20,...
24	VY2FL	8E10.3	1 - 10,11 - 20,...
25	IBOBC	80I1	1...80
26	VBOCH	8F10.0	1 - 10,11 - 20,...
27	VBOFL	8E10.3	1 - 10,11 - 20,...
28	N1	I5	1 - 5,
	N2	I5	6 - 10
29	I	I5	1 - 5,
	COX	E10.3	6 - 15
30	ICOX	80I1	1...80
31	I	I5	1 - 5,
	COY	E10.3	6 - 15
32	ICOY	80I1	1...80
33	I	I5	1 - 5,
	COZ	E10.3	6 - 15
34	ICOZ	80I1	1...80
35	I	I5	1 - 5,
	STO	E10.3	6 - 15
36	ISTO	80I1	1...80
37	I	I5	1 - 5,
	POR	E10.3	6 - 15
38	IPOR	80I1	1...80
39	I	I5	1 - 5,
	DISX	E10.3	6 - 15,
	DISY	E10.3	16 - 25,
	DISZ	E10.3	26 - 35,
	DIFU	E10.3	36 - 45
40	ITU	I5	1 - 80
41	T1	E10.3	1 - 10,
	T2	E10.3	11 - 20,
	NPP	I5	21 - 25,
	NW	I5	26 - 30
42	IDF	I5	1 - 5,
	TFLUX	E10.3	6 - 15
43	ISIDE	I5	1 - 5,
	ARE	4F10.3	6 - 15,16 -25,...
44	NTR	I5	1 - 5,

	NSKIP	I5	6 - 10,
	DELR	E10.3	11 - 21
45	NPART	16I5	1 - 5,6 -10,...

SPECIAL CARDS

Internal Nodes Specifications
(IF NICH or NIFL > 0)

9A	IDICH	I5	1 - 5,
	VICH	E10.3	6 - 15
9B	IDIFL	I5	1 - 5,
	VIFL	E10.3	6 - 15

Sink Specifications
(IF NSINK > 0)

27A	N1	I5	1 - 5,
	N2	I5	6 - 10

3.3 Dimensioning of the Arrays

There are several subscripted variables used in the program which must be properly dimensioned. Such changes are only required in a separate common block file since the proper segments of this file, containing the dimension statements and common blocks, are accessed directly from the program segments in which they are used through the use of "$CONTINUE ..." statements. The array dimensioning parameters are the following:

NTOP	: number of constant head nodes at the TOP boundary
NX1	: number of constant head nodes at the X1 boundary
NX2	: number of constant head nodes at the X2 boundary
NY1	: number of constant head nodes at the Y1 boundary
NY2	: number of constant head nodes at the Y2 boundary
NBOT	: number of constant head nodes at BOTTOM boundary
NICH	: number of internal constant head nodes
NIFL	: number of internal specified flux nodes
NODX	: number of nodes in the X-direction
NODY	: number of nodes in the Y-direction
NODZ	: number of nodes in the Z-direction
NXEL	: number of elements in the X-direction
NYEL	: number of elements in the Y-direction

```
NZEL    : number of elements in the Z-direction
NCOX    : number of X-direction conductivity units
NCOY    : number of Y-direction conductivity units
NCOZ    : number of Z-direction conductivity units
NSTO    : number of storage coefficient units
NPOR    : number of porosity units
NTU     : number of transport units
NW      : number of transient flux nodes
NNODE   : total number of nodes
NEL     : total number of elements
NCHEAD  : total number of constant head nodes
NONUL   : dimension of coefficient storage array (NF * 14)
NF      : number of degrees of freedom (NNODE - NCHEAD)
NFLTOP  : number of specified flux nodes at the TOP boundary
NFLX1   : number of specified flux nodes at the X1 boundary
NFLX2   : number of specified flux nodes at the X2 boundary
NFLY1   : number of specified flux nodes at the Y1 boundary
NFLY2   : number of specified flux nodes at the Y2 boundary
NFLBOT  : number of specified flux nodes at BOTTOM boundary
NIFL    : number of internal specified flux nodes
NTOT    : maximum number of particles in the flow system
TRSTEPS: number of transport steps
NFL     : total number of flux nodes
NSP     : workspace dimension  (See information below)
```

The workspace is an array used by the Cholesky sparse matrix solver. Its storage requirements varies greatly as a function of the finite element grid size. The critical parameters are the number of degrees of freedom (NF) and the bandwidth (NB). For relatively large problems the total storage requirements can be approximated by:

```
NSP = .1 *(NB * NF**1.4)
```

- for regular grid NB = (NODZ*(NODY+1))+2
- for a modified grid NB is less

In any cases the following relationship should hold:

```
NSP > 32 * NF
```

The value of NSP used in the datafile HAS TO BE identical to that used for the array dimensioning otherwise protection exception. If NSP is too small the program will return a non-zero FLAG from the sparse matrix solver thus indicating that NSP should be increased. When a proper dimensioning is used, FLAG=0 in the output file. The upper limit for the workspace, linked to the nature of our hardware support, is by default of one(1) Megabyte or 131072 double precision entries (i.e. NSP can not be greater than 131072).

```
example:   NF = 366
           NB = 123     ->    366**1.4 = 3880

           NSP = .1 *(123 * 3880)
               = 47729

also       NSP > 50 * NF = 11712
```

An estimate of the CPU execution time T(SEC) can be obtained by a similar expression.

```
example:   T = (NB * NF)/4000  SEC

           T = 11.25 SEC
```

Knowing the cost per page-minute and the priority rate, a cost estimate can be obtained:

```
example:   NSP = (47729 * 8) or 381832 bytes
               = 373 pages   (1 page = 1Kbyte)

           CPU usage:
                 (373 * 11.25) page-minutes
                 or  69.9 p-m

           COST = (Priority rate (per p-m) * 69.9) $$$
```

Following is a listing of the common block file to be edited so as to dimension explicitly each array.

```
      DIMENSION VTOCH( NTOP ),IOP(30),IDWT(  NODX , NODY ),ITOBC(  NODX , NODY ),
     *          VX2CH( NX2 ),IX1BC( NODY , NODZ ),VX1CH( NX1 ),IX2BC( NODY , NODZ ),
     *          VY2CH( NY2 ),IY1BC( NODX , NODZ ),VY1CH( NY1 ),IY2BC( NODX , NODZ ),
     *          VBOCH( NBOT ),IBOBC( NODX , NODY ),VICH( NICH ),VIFL( NIFL )
      DIMENSION ICOX( NXEL , NYEL , NZEL ),ICOY( NXEL , NYEL , NZEL ),ICOZ( NXEL , NYEL , NZEL ),
     *          IDICH( NICH ),IDIFL( NIFL ),ISTO( NXEL , NYEL , NZEL ),
     *          IPOR( NXEL , NYEL , NZEL ),POR( NPOR ),
     *          COX( NCOX ),COY( NCOY ),COZ( NCOZ ),STO( NSTO )
      DIMENSION ITOCX( NTOP ),ITOCY( NTOP ),ITOCZ( NTOP ),
     *          IBOCX( NBOT ),IBOCY( NBOT ),IBOCZ( NBOT )
      DIMENSION IX1CX( NX1 ),IX1CY( NX1 ),IX1CZ( NX1 ),
     *          IX2CX( NX2 ),IX2CY( NX2 ),IX2CZ( NX2 )
      DIMENSION IY1CX( NY1 ),IY1CY( NY1 ),IY1CZ( NY1 ),
     *          IY2CX( NY2 ),IY2CY( NY2 ),IY2CZ( NY2 )
      DIMENSION INCHX( NICH ),INCHY( NICH ),INCHZ( NICH ),
     *          INFLX( NIFL ),INFLY( NIFL ),INFLZ( NIFL ),IDF( NW )
      REAL*8 TD
C
      COMMON / IO / IN,IOUT
      COMMON / OPTION / IOP
      COMMON / MESH1 / VOL,DELX,DELY,DELZ,NODX,NODY,NODZ,
     1                 NCOX,NCOY,NCOZ,NB,NBE,NF,NSTO,NPOR
      COMMON / BOUND / VTOCH,VX1CH,VX2CH,VY1CH,VY2CH,VBOCH,VICH,VIFL,
     1             IDWT,ITOBC,IX1BC,IX2BC,IY1BC,IY2BC,IBOBC,IDICH,
     2             NTOP,NX1,NX2,NY1,NY2,NBOT,NCHEAD,NICH,NIFL,IDIFL
      COMMON / GEO / STO,POR,COX,COY,COZ,ICOX,ICOY,ICOZ,NXEL,NYEL,NZEL
     *     ,ISTO,IPOR
      COMMON / CH / ITOCX,ITOCY,ITOCZ,IBOCX,IBOCY,IBOCZ,
     1             IX1CX,IX1CY,IX1CZ,IX2CX,IX2CY,IX2CZ,
     2             IY1CX,IY1CY,IY1CZ,IY2CX,IY2CY,IY2CZ,
     3             INCHX,INCHY,INCHZ,INFLX,INFLY,INFLZ
      COMMON / TIME / TD,DELT,T,T1,T2,IDF,ISTEP,NPP,NW
C     ------------------------------------------------------
      REAL*8 X( NNODE ),Y( NNODE ),Z( NNODE ),CX( NEL ),CY( NEL ),CZ( NEL ),
     *       CONST( NCHEAD ),S( NEL ),PORO( NEL )
      INTEGER*4 IM( NODZ , NODY , NODX ),INC(8, NEL ),ITYPE( NNODE ),NEL,NNODE
C
      COMMON / GRID1 / S,PORO,X,Y,Z,CX,CY,CZ,CONST,IM,INC,ITYPE,NEL
     *     ,NNODE
C     ------------------------------------------------------
      REAL*8 COOR(2),NS(8),NXSI(8),NETA(8),NZET(8),
     1       DNX(8),DNY(8),DNZ(8),
     2       NX( NEL ,8),NY( NEL ,8),NZ( NEL ,8),GE(8,8),GEL,PE(8,8)
      INTEGER INEL(8),IORD(8)
C
      COMMON / QUAD1 / NS,NXSI,NETA,NZET,DNX,DNY,DNZ,
     1                 NX,NY,NZ,GE,INEL,IORD
      COMMON / QUAD2 / COOR
C     ------------------------------------------------------
      REAL* 8 PP( NONUL ),PHF( NF ),FCH( NF ),FBC( NF ),
```

```
     *  ZHEAD( NNODE ),HEAD( NF )
      INTEGER LC( NNODE )
C
      COMMON /SOL1 / PP,FBC,PHF,FCH,ZHEAD,HEAD,LC,NEW,NSS
C     ----------------------------------------------------------
      REAL*8 A( NONUL ),RSP( NSP ),AA( NONUL )
C
      INTEGER FLAG
      INTEGER*4 IA( NF+1 ),JA( NONUL ),PO( NF ),IP( NF ),PATH,
     *        ISP( NSP ),ESP,IPATCH(14),IUN(14),ICOR(8),NSP,
     *        IPP( NF+1 ),JPP( NONUL )
      EQUIVALENCE(ISP(1),RSP(1))
      COMMON / YALE1 / A,AA,PO,IPATCH,INI,IA,JA,JPP,NSP
C     ----------------------------------------------------------
      REAL*8    VTOFL( NFLTOP ),VX1FL( NFLX1 ),VX2FL( NFLX2 ),FLUX( NFL ),
     *          VBOFL( NFLBOT ),VY1FL( NFLY1 ),VY2FL( NFLY2 ),TFLUX( NW )
      DIMENSION ITOFX( NFLTOP ),ITOFY( NFLTOP ),ITOFZ( NFLTOP ),
     *          IBOFX( NFLBOT ),IBOFY( NFLBOT ),IBOFZ( NFLBOT )
      DIMENSION IX1FX( NFLX1 ),IX1FY( NFLX1 ),IX1FZ( NFLX1 ),
     *          IX2FX( NFLX2 ),IX2FY( NFLX2 ),IX2FZ( NFLX2 )
      DIMENSION IY1FX( NFLY1 ),IY1FY( NFLY1 ),IY1FZ( NFLY1 ),
     *          IY2FX( NFLY2 ),IY2FY( NFLY2 ),IY2FZ( NFLY2 )
C
      COMMON / BOUNDF / NFLUX,NFLTOP,NFLBOT,NFLX1,NFLX2,
     *       NFLY1,NFLY2,NFL
      COMMON / FX / VTOFL,VX1FL,VX2FL,VBOFL,VY1FL,VY2FL,FLUX,TFLUX
      COMMON / GRID2 / ITOFX,ITOFY,ITOFZ,IBOFX,IBOFY,IBOFZ,
     *                 IX1FX,IX1FY,IX1FZ,IX2FX,IX2FY,IX2FZ,
     *                 IY1FX,IY1FY,IY1FZ,IY2FX,IY2FY,IY2FZ
C     ----------------------------------------------------------
      REAL*8 VDIR(3,8),PCOR(3),VLEV1(3),VLEV2(3),DGRID(3)
      REAL*8 ARE(4),XM( NTOT ),YM( NTOT ),ZM( NTOT ),XVN,YVN,ZVN,DELR,
     *   TRT,XMI,YMI,ZMI,XLIM,YLIM,ZLIM
      REAL*8 DX,DY,DZ,VPX,VPY,VPZ,PMAS( NTOT ),SMAS( NNODE ),PINMAS
      REAL*8 XV,YV,ZV,V1,V2,V3
      DIMENSION IMESH(3),NR(3),ILOC( NZEL , NYEL , NXEL ),
     *    CONC( NZEL , NYEL , NXEL )
      DIMENSION ISINK( NNODE )
      INTEGER NPART( TRSTEPS ),ITU( NXEL , NYEL , NZEL )
      DIMENSION DISX( NTU ),DISY( NTU ),DISZ( NTU ),DIFU( NTU )
C
      COMMON / MASS1 / VDIR,VPX,VPY,VPZ,DELR,ILOC,IDP
      COMMON / TRANS / ARE,XM,YM,ZM,TCUM,ISIDE,NPART,NTR,NST,NTOT,KT,
     *     NSKIP
      COMMON / PROBE / IRX
      COMMON / PMT / PMAS,SMAS,PINMAS,XMI,YMI,ZMI,NPR,NSKTOT
      COMMON / PRSNK / ISINK,NSINK
      COMMON / VELOP / XV,YV,ZV,V1,V2,V3,X1,Y1,Z1,RES
      COMMON / MASS2 / XLIM,YLIM,ZLIM,DISX,DISY,DISZ,DIFU,NTU,ITU,IPAR
      COMMON / MASS3 / DX,DY,DZ,DISPX,DISPY,DISPZ,DIF
```

4. MODE OF OPERATION

The purpose of this section is to illustrate the use of the program with an application. In order to use the model it is first required to prepare the data file according to the instructions given in the previous sections. The dimensioning requirements can then be figured out, followed by the actual dimensioning of the arrays, the compilation of the program and the simulation run.

4.1 Array Dimensioning

1. Make a copy of the template common block file COMMON.USER to be used as the array dimensioning file,
   e.g. COPY COMMON.USER TO BLOCK@I
   (the use of the command modifier "@I" is necessary in order to preserve an Identical line numbering)

2. Edit common bloc file by replacing each parameter name by the appropriate integer
   EDIT BLOCK
   e.g. A@A /F # NCHEAD #120#
   (the parameter name MUST be included between blanks)
   * every parameter must receive a value
   * the smallest value is 1, otherwise compiling or execution failure will occur.
   * the dimensions should be at least as large as the maximal storage requirement

3. Make a copy of the SOURCE CODE: 3DFT
   e.g. $SET IC=OFF
   $COPY 3DFT TO SOURCE@I
   $SET IC=ON
   (Implicit Concatenation MUST BE disabled prior to the file copy, otherwise the proper common block file will not be accessed at compilation time)

4. Edit the source code "$CONTINUE" statements so as to branch to the proper common block file (e.g. BLOCK)
   e.g. statements should be:
   $CONTINUE WITH BLOCK(1,99) RETURN
   $CONTINUE WITH BLOCK(100,199) RETURN
   .
   .
   .

These equations insure that at compilation time the proper sections of the common block file are 'Implicitely Concatenated' (IC=ON), i.e. read and compiled as part of the

various program segments. This procedure unables the user to modify the array dimensioning safely and consistently, by editing only once.

4.2 Compilation

The Amdahl 5860 allows to compile, task build and link at once automatically by using the following command:

```
$RUN *FORTG SCARDS=Sourcecode O=Object
```

The code is equally compatible with the FORTGTEST and FORTX compiler and can be debugged using the INTERACTIVE FORTRAN interpreter *IF.

4.3 Simulation Run

The simulation is run by issuing the following command:

```
$RUN Object 5=Datafile 6=Output 7=Scratchfile T=Time
```